

AN OPTIMAL NETWORK-BASED APPROACH TO SCHEDULING AND
RE-ROSTERING CONTINUOUS HETEROGENEOUS WORKFORCES

by

Shane A. Knighton

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

August 2005

| Report Documentation Page | | | | Form Approved OMB No. 0704-0188 | |
|--|------------------------------------|-------------------------------------|--|---|------------------------------------|
| Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. | | | | | |
| 1. REPORT DATE 01 AUG 2005 | | 2. REPORT TYPE N/A | | 3. DATES COVERED - | |
| 4. TITLE AND SUBTITLE An Optimal Network-Based Approach To Scheduling And Re-Rostering Continuous Heterogeneous Workforces | | | | 5a. CONTRACT NUMBER | |
| | | | | 5b. GRANT NUMBER | |
| | | | | 5c. PROGRAM ELEMENT NUMBER | |
| 6. AUTHOR(S) | | | | 5d. PROJECT NUMBER | |
| | | | | 5e. TASK NUMBER | |
| | | | | 5f. WORK UNIT NUMBER | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Arizona State University | | | | 8. PERFORMING ORGANIZATION REPORT NUMBER | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) The Department of the Air Force AFIT/CIA WPAFB, OH | | | | 10. SPONSOR/MONITOR'S ACRONYM(S) | |
| | | | | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) | |
| 12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited | | | | | |
| 13. SUPPLEMENTARY NOTES The original document contains color images. | | | | | |
| 14. ABSTRACT | | | | | |
| 15. SUBJECT TERMS | | | | | |
| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT UU | 18. NUMBER OF PAGES 197 | 19a. NAME OF RESPONSIBLE PERSON |
| a. REPORT unclassified | b. ABSTRACT unclassified | c. THIS PAGE unclassified | | | |

AN OPTIMAL NETWORK-BASED APPROACH TO SCHEDULING AND
RE-ROSTERING CONTINUOUS HETEROGENEOUS WORKFORCES

by

Shane A. Knighton

has been approved

July 2005

APPROVED:

_____, Chair

Supervisory Committee

ACCEPTED:

Department Chair

Dean, Division of Graduate Studies

ABSTRACT

This dissertation presents a mathematical programming approach to the personnel tour scheduling problem based upon a minimum cost network flow formulation with specialized side constraints. The linear program optimally solves the tour scheduling problem for industries with continuous (24 hour) operations and a heterogeneous workforce such as fast food restaurants, hotels and resorts, university computer labs, nurses in large health care systems, and retail sales. The methodology allows for a workforce with varying availabilities, part and full-time employees, differing skill-sets and wage rates, and minimum and maximum shift requirements per week. Additionally, the formulation can quickly adjust an optimal schedule due to sickness, vacation, hiring or firing with the workforce, and minimize the number of deviations within the original schedule. A methodology that can optimally schedule both a continuous and a heterogeneous workforce in an insignificant amount of computational time is unique in the literature. In many instances the linear program generates integral solutions to the tour-scheduling problem without branching, bounding, or cutting techniques. An interior-point method solves the formulation in less than 3 seconds for large problem instances of 80 employees scheduled to 420 distinct shifts. The efficiency of the formulation is presented for many test sets of problems as well as an application to the tour-scheduling problem of computer lab staffing at Arizona State University and nurses with the Banner Health network of health care providers.

DEDICATION

This accomplishment is a direct result of the unyielding love and support of my wife. You have made my life complete and are my motivation to succeed. It is all impossible without you. To my friends whose support was incredible and to my family whose love is unwavering, I love you and I thank you. I love you too Quila.

Thank you to the United States Air Force for the wonderful opportunity to pursue the degree.

ACKNOWLEDGMENTS

I want to express my extreme gratitude to my committee chair, Dr. Jeffery Cochran, for his insight, support and most importantly, his guidance. Thank you to the remaining members of my committee for their oversight and help.

Additionally, I would like to thank Maj Tom Boushell for his assistance and friendship. Thank you to Tom and Jeff Laub for their help in sculpting the ASU Phase 1 computer lab problem and to James Broyles for his help in scoping the nurse scheduling problem at Banner Health.

Thank you to the faculty and staff of the Industrial Engineering Department at Arizona State University. I have learned a tremendous amount from all of you.

The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

TABLE OF CONTENTS

| CHAPTER | Page |
|--|------|
| LIST OF TABLES | ix |
| LIST OF FIGURES | x |
| 1 INTRODUCTION | 1 |
| 1.1 Overview | 1 |
| 1.2 Statement of the Problem | 5 |
| 1.3 Original Contribution | 6 |
| 1.4 Organization | 7 |
| 2 LITERATURE REVIEW | 8 |
| 2.1 Overview | 8 |
| 2.2 Problem Instance Classification | 9 |
| 2.3 Application Areas | 12 |
| 2.4 Solution Techniques and Methodologies | 15 |
| 2.5 Linear Programming Approach to Tour Scheduling | 25 |
| 2.6 Literature Summary | 27 |
| 3 A NETWORK-FLOW BASED MATHEMATICAL PROGRAMMING APPROACH TO OPTIMAL ROSTERING OF CONTINUOUS HETEROGENEOUS WORKFORCES | 29 |
| Abstract | 29 |
| 3.1 Introduction | 30 |
| 3.2 Literature Survey | 35 |

| CHAPTER | Page |
|---|------|
| 3.3 A Case Study – ASU Computer Lab Technician Tour Scheduling | 40 |
| 3.4 Proposed Tour-Schedule Solution Method and Model | 45 |
| 3.5 Computational Results..... | 52 |
| 3.6 Conclusions and Recommendations | 58 |
| 4 A MATHEMATICAL PROGRAMMING APPROACH TO OPTIMAL NURSE ROSTERING OVER CONTINUOUS AND MULTI-WEEKLY TIME PERIODS WITH EMPLOYEE WEEKEND RESTRICTIONS AND SHIFT PREFERENCES | 60 |
| Abstract..... | 60 |
| 4.1 Introduction | 61 |
| 4.2 Literature Survey | 64 |
| 4.3 Nurse Tour-Scheduling Problem..... | 66 |
| 4.4 Proposed Tour-Schedule Solution Method and Model | 67 |
| 4.5 Computational Results..... | 78 |
| 4.6 Conclusions | 83 |
| 5 A NETWORK-BASED MATHEMATICAL PROGRAMMING APPROACH TO USING EMPLOYEE PREFERENCES IN RE-ROSTERING OPTIMAL TOUR-SCHEDULES | 84 |
| Abstract..... | 84 |
| 5.1 Introduction | 85 |
| 5.2 Literature Summary | 87 |
| 5.3 Re-rostering Case Studies..... | 91 |
| 5.4 Proposed Tour-Schedule Solution Method and Model | 93 |

| CHAPTER | Page |
|--|------|
| 5.5 Re-rostering Results | 98 |
| 5.6 Conclusion | 99 |
| 6 CONTRIBUTIONS AND FUTURE RESEARCH | 101 |
| 6.1 Contributions..... | 101 |
| 6.2 Future Research | 104 |
| REFERENCES..... | 110 |
| APPENDIX A CHAPTER THREE VISUAL BASIC CODE | 114 |
| APPENDIX B CHAPTER THREE CPLEX OUTPUT | 123 |
| APPENDIX C CHAPTER FOUR VISUAL BASIC CODE | 139 |
| APPENDIX D CHAPTER FOUR CPLEX OUTPUT | 160 |
| APPENDIX E CHAPTER FIVE VISUAL BASIC CODE..... | 170 |
| APPENDIX F CHAPTER FIVE CPLEX OUTPUT | 174 |

LIST OF TABLES

| TABLE | Page |
|--|------|
| 2.1 Number of surveyed papers in each application area of personnel scheduling | 13 |
| 2.2 Number of surveyed using each methodology for personnel scheduling | 16 |
| 2.3 Number of papers reviewed that specifically addressed the tour-scheduling problem instance using the methodology breakdown and application areas proposed by Ernst, et. al. | 28 |
| 3.1 Lab technician requirements (Atrium) based on hourly demand | 42 |
| 3.2 Shift penalties used in Phase 1 set covering problem..... | 44 |
| 3.3 Results of the tour-schedule model for ASU computer labs | 53 |
| 3.4 Computation results for larger problem instances | 55 |
| 3.5 Employee requirements as a function of employee availability | 57 |
| 4.1 Summary of results for formulation with and without IP variables | 79 |
| 4.2 Computational time results..... | 81 |
| 5.1 Re-rostering results..... | 99 |

LIST OF FIGURES

| FIGURE | Page |
|--|------|
| 1.1 Number of employees required at hourly intervals is the result of demand modeling..... | 3 |
| 1.2 Set of shift to optimally cover customer demand from Figure 1.1..... | 4 |
| 2.1 Multicommodity flow problem in a directed multi-level acyclic network . | 20 |
| 3.1 Employees requirement by hour from demand modeling..... | 32 |
| 3.2 Example set of shifts that cover the demand in Fig. 3.1 | 33 |
| 3.3 Two-phase methodology for shift selection and rostering | 43 |
| 3.4 Example of effect of preference on two distinct shift definitions | 44 |
| 3.5 Network representation of the Phase 2 tour-scheduling problem..... | 46 |
| 3.6 Input data file..... | 51 |
| 3.7 Computational time comparison versus the number of variables | 55 |
| 3.8 Computational time comparison versus the number of constraints | 56 |
| 3.9 Employee to shift ratio based upon employee availability | 58 |
| 4.1 Network representation of the Phase 2 tour-scheduling problem..... | 69 |
| 4.2 Multi-weekly network representation of tour-scheduling problem..... | 71 |
| 4.3 Illustration of aggregate preferences for employee/shift combination .. | 76 |
| 4.4 Employee/Manager shift preferences input data file | 77 |
| 4.5 List of conflicted shifts input file..... | 78 |
| 4.6 CPLEX header output | 80 |
| 4.7 Comparison of computational times and number of constraints..... | 82 |

| FIGURE | Page |
|--|------|
| 4.8 Comparison of computational time and number of variables | 82 |
| 5.1 Multicommodity flow problem in a directed multi-level acyclic network . | 89 |
| 5.2 Illustration of aggregate preferences for employee/shift combination ... | 95 |
| 5.3 Preferences selected in original optimal schedule | 96 |
| 5.4 Absence of employee 12..... | 97 |
| 5.5 Modified preference set used in re-rostering..... | 98 |
| 6.1 Network-flow representation of combined problem | 106 |
| 6.2 Combined problem with varied shift lengths..... | 108 |

Chapter 1

INTRODUCTION

1.1 Overview

Personnel scheduling is one of the most difficult, important, and often studied problems in operations research. The optimal choice of the number of employees required to meet customer demand, shift start and stop times, daily lunches and breaks, and the assignment of the adequately skilled employee to the best shift is a large combinatorial optimization problem. Managers who attempt to manually solve the personnel scheduling problem expend many valuable work hours to find even a feasible solution which has little probability of being optimal based on any objective function. Sub-optimal scheduling increases an industry's tangible costs, not only through the consumption of a manager's time, but also through the misallocation of shifts to meet customer demand and through employees staffing shifts for which they are not qualified. Furthermore, a sub-optimal schedule will increase intangible costs such as lower employee morale from dissatisfaction with a poor work schedule. The problem is further complicated by employee sicknesses, vacations, hiring, or firing. In this inevitable instance the sub-optimal problem must again be adjusted, costing more hours, to find another feasible schedule that will also likely be sub-optimal. Eliminating these costs is an easy way to increase the profitability of any business.

Personnel scheduling is a necessary chore associated with any organization employing people to get work accomplished. Some organizations

that employ a professional workforce during traditionally scheduled business hours find the personnel scheduling problem a trivial concern. However, industries that rely on a large portion of part-time employees with highly constrained availabilities (work hours), varying skill sets within the workforce, and require staffing for the entire day and evening hours or even continuous (24-hour) operations have a much more challenging problem. A timely and optimal solution to the personnel scheduling problem will save each manager valuable time, meet customer demand, and increase worker morale by matching each employee to the best shift possible. This dissertation presents an extremely fast, computationally efficient, and optimal network-flow based mathematical programming solution for the extremely challenging problem of rostering and re-rostering a continuous heterogeneous workforce.

The large, complex problem of personnel scheduling is generally decomposed into more tractable sub-problems. The three primary sub-problems are demand modeling, shift selection, and employee tour-scheduling or rostering. Demand modeling determines the number of employees required during a given time interval to satisfy customer needs. Forecasting and queuing theory are the primary tools applied to demand modeling. A typical chart of the number of employees required, illustrating the results of demand modeling, is shown in Figure 1.1.

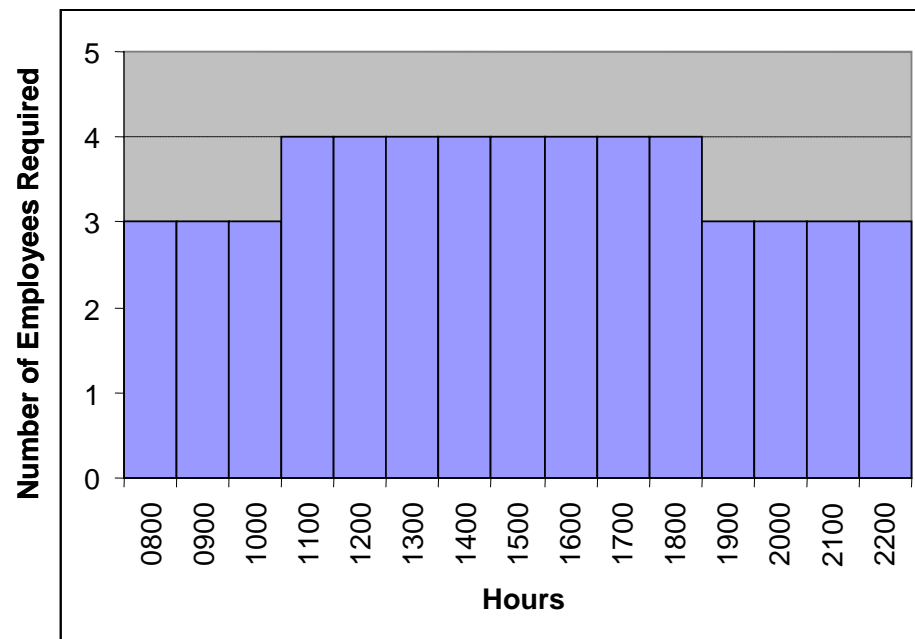


Figure 1.1: Number of employees required at hourly intervals is the result of demand modeling

The second primary personnel scheduling sub-problem is shift selection. Shift selection assigns consecutive hour shifts to satisfy the customer demand while meeting organizational and regulatory requirements for shift lengths, breaks, and mealtime allowances. The objective of shift selection is generally to minimize the number of hours scheduled that exceed customer demand. Figure 1.2 shows an example of a set of shifts that will optimally cover the number of employees required in Figure 1.1.

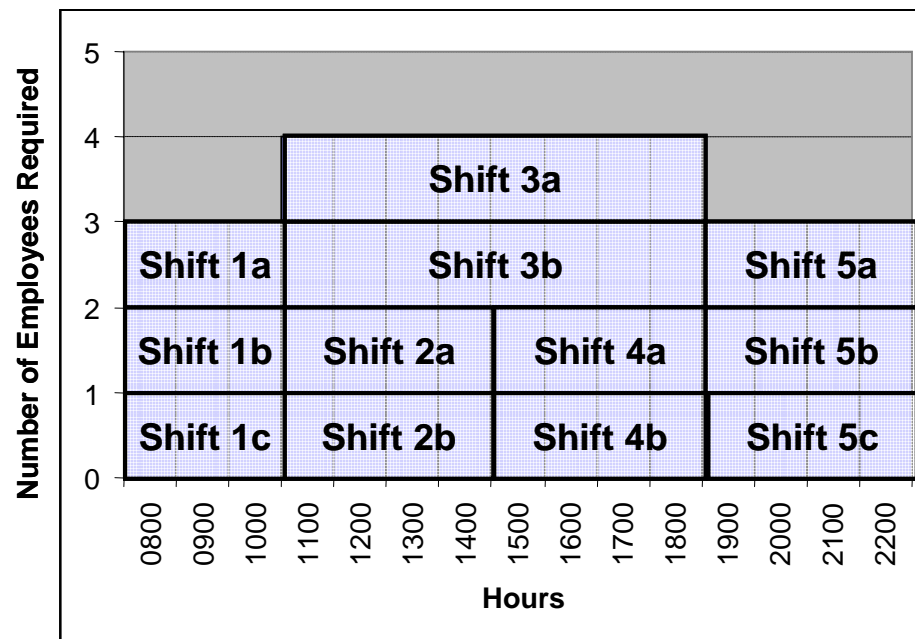


Figure 1.2: Set of shifts to optimally cover customer demand from Figure 1.1

Finally, the third primary personnel scheduling sub-problem is tour-scheduling also known as employee rostering. Tour-scheduling assigns individuals to specific shifts detailed during shift selection. Tour-scheduling problem complexity differs in the literature depending upon the nature of the workforce under consideration. This dissertation will look at one of the most complex tour-scheduling problems due to its consideration of a heterogeneous continuous workforce that gives consideration to employee shift preferences and managerial employee preferences. A heterogeneous workforce is a collection of personnel who have significantly variant availabilities, skill sets, and wage rates. Tour-scheduling a heterogeneous workforce must consider all of these differences and optimally match the personnel with the best shift possible and for which they are eligible. Additionally, considerations for individuals working a

minimum and maximum number of shifts per week and insuring adequate rest between shifts for industries with continuous (24-hour) operations are critical to an optimal tour schedule. Employee shift preferences and management employee weighting comprise the objective function.

1.2 Statement of the Problem

This dissertation will optimally solve the tour-scheduling problem for a continuous heterogeneous workforce, meeting all minimum and maximum shift regulations, skill set requirements, employee availabilities, and rest periods between shifts. The formulation proposed will provide a solution to the tour-scheduling problem in an insignificant amount of computational time for large problem instances. A set of test problems are given to benchmark the formulation proposed in this dissertation. Optimally solving the tour-scheduling sub-problem does not guarantee a global optimal solution to the combined shift selection/tour-scheduling problem. An alternate set of shifts that minimizes excess employee hours, but increases the number of preferred shifts of personnel can only be found by solving the combined shift selection tour-scheduling problem. The proposed methodology is examined to solve the combined problem.

1.3 Original Contribution

The work in this dissertation will provide several original contributions to the personnel scheduling literature. The following list summarizes the advancements proposed by this research.

- Provide a solution methodology that can simultaneously account for variations in employee availabilities, skill sets, and wage rates
- Provide a methodology that can roster employees for continuous (24-hour) operations by allowing for rest hours between shifts
- Develop a network-flow based formulation that will rapidly solve the tour-scheduling problem, providing integral answers without branching, bounding, or cutting techniques
- Roster beyond the traditional weekly schedules to monthly, quarterly, and yearly schedules while restricting the amount of weekends worked when creating multi-week tour schedules
- Provide a framework that only requires a minimal adjustment to the formulation to re-roster the workforce when perturbations in an optimal schedule occur due to employee sickness, vacations, hiring, or firing
- Allow for each employee to provide preferences for more favorable shifts and each manager to provide preference for employees based on seniority or performance

- Explore the ability to expand the network-flow formulation to simultaneously solve both the shift selection and tour-scheduling problem for a continuous heterogeneous workforce

1.4 Organization

This dissertation is organized in the following manner. Chapter 2 contains a survey of the pertinent literature and important previous work accomplished in personnel scheduling. Chapters 3, 4, and 5 each contain a complete stand alone journal article which, when taken together, embody the research and original contributions set forth in this dissertation. Chapter 3 is a journal article entitled “A Network-based Mathematical Programming Approach to Optimal Rostering of Continuous Heterogeneous Workforces” detailing a mathematical programming formulation for the tour-scheduling problem. Chapter 4 expands the model in Chapter 3 to multi-weekly time horizons and restricts the number of weekends worked by each employee. The article is entitled “A Mathematical Programming Approach to Optimal Nurse Rostering over Continuous and Multi-Weekly Time Periods with Employee Weekend Restrictions and Shift Preferences”. Chapter 5 contains an article entitled “A Network-based Mathematical Programming Approach to Using Employee Preferences in Re-rostering Optimal Tour-Schedules”. This work modifies the models presented in Chapters 3 and 4 to provide a mechanism for re-rostering an optimal schedule that has been perturbed by employee absences. Chapter 6 presents the overall conclusions and recommendations. Appendices follow the references.

Chapter 2

LITERATURE SURVEY

2.1 Overview

A vast amount of scholastic work has been accomplished on the personnel scheduling problem. This work has been documented in a wide array of refereed journals, conference proceedings, and lecture notes. Personnel scheduling spans a wide range of problem instances, specific applications, and solution techniques and methodologies.

The journal *Annals of Operations Research* published two special issues in March and April 2004 dedicated to the personnel scheduling problem. The issues are comprised of 22 personnel scheduling papers of which “2 are survey papers, 18 are applications-oriented papers in specific application areas, and 2 talk to general methodological and theoretical advances” (Jiang et. al., 2004). These special issues indicate the continued interest in the personnel scheduling problem as well as the breadth of research topics being explored by the research community. This literature review will make extensive use of the articles in these special issues in addition to other prior pertinent work in personnel scheduling.

This literature review will begin by defining specific problem instances studied in the personnel scheduling literature. The next section will describe many of the application areas using personnel scheduling and the solution techniques and methodology applied. Finally, the literature review will concentrate on the prior work dedicated to the tour-scheduling problem using network flow problems for an industry characterized by flexible demand, a

heterogeneous workforce, and continuous operations. The relatively small quantity of work in this area is contrasted with the proposed methodology of this dissertation.

2.2 Problem Instance Classification

Classification of the problem instances, application areas, and solution techniques is essential to understanding the focus of the work proposed in this dissertation. Ernst, et. al. (2004a) propose such definitions for classification of the literature in personnel scheduling. Of primary initial concern are the definitions of the problem instances currently under study. In their article, Ernst et. al. define six problem instances. They include: demand modeling, days off scheduling, shift scheduling, line of work construction, task assignment, and staff assignment.

2.2.1 Demand Modeling

Demand modeling is the first step in the process of personnel scheduling. Demand modeling concentrates on the determination of “how many staff are needed at different times over some planning period. Demand modeling is the process of translating a predicted pattern of incidents into associated duties and then using the duty requirements to ascertain a demand for staff” (Ernst et. al., 2004a). Demand modeling is further broken down into three distinct types.

The first subcategory is task based demand. Task based demand “is obtained from lists of individual tasks to be performed” (Ernst et. al., 2004a). The tasks are of a specified length and may have preset start times at which the task

must begin. Applications include airline flight schedules or public transportation. Scheduling to accommodate task based demand is often “associated with crew pairing generation and crew pairing optimization” (Ernst et. al., 2004a).

The second subcategory is flexible demand. Flexible demand is characterized by customer service requirements that are not fixed and cannot be exactly known. In general, this demand is measured using stochastic modeling or forecasting; the uncertain demand is predicted and staffing levels are determined. Application areas are associated with customer service driven industries such as call-centers, financial institutions, and restaurants.

Finally, the third subcategory defined by Ernst, et. al. (2004a) is shift based demand. Shift based demand is characterized by a minimum number of staff required at specific time intervals. “Shift based demand arises naturally in applications such as nurse scheduling and ambulance services where staff levels are determined from a need to meet service measures such as nurse/patient ratios or response times” (Ernst et. al., 2004a). It is pointed out that shift based demand can be considered a simplification of both task based and flexible demand problem instances.

2.2.2 Days off Scheduling

Once the demand has been determined, the next step in personnel scheduling may be to determine the associated rest periods for a specific service organization. Examples might be operating hours of a bank, not operating a garage on the weekends, or a fast-food restaurant remaining open later on the weekends. This analysis is often characterized by a cost analysis based upon

the predicted demand and is predominantly used when scheduling “flexible or shift based demand rather than when rostering to task based demand” (Ernst et. al., 2004a).

2.2.3 Shift Scheduling

Shift scheduling chooses the appropriate start and finish times for a particular shift. The overall set of shifts will provide for the required number of personnel to meet the demand required or predicted through demand modeling. The overall goal is “to select a good set of feasible duties, shifts or (crew) pairings to cover all tasks” or demand (Ernst et. al., 2004a). Shift scheduling must incorporate such constraints as maximum and minimum shift lengths prescribed by organizational policy or by law and may specify break and meal periods.

2.2.4 Line of Work Construction

Line of work construction assigns personnel to the set of shifts found through days off scheduling and shift scheduling. Personnel availability, skill set, and preferences are considerations when assigning shifts. Additionally, shift conflicts and patterns are important, “for example it might not be possible to immediately follow a sequence of night shifts with a day shift” (Ernst et. al., 2004a). Ernst, et. al. point out that line of work construction is generally called “tour-scheduling” or “rostering” for flexible demand industries and “crew rostering” for task based demand. This dissertation will concentrate on the tour-scheduling problem.

2.2.5 Task and Shift Assignment

Task Assignment and Shift Assignment are considered by some in the literature to be redundant problem instances, but are presented here for the purpose of completeness. Task Assignment assigns personnel, previously scheduled to a particular shift, a set of specific tasks to be completed during that shift. Shift assignment “involves assignment of individual staff to the lines of work” and “is often done during the construction of the work lines” (Ernst et. al., 2004a).

2.3 Application Areas

Personnel scheduling has a broad range of application. In a separate article Ernst, Jiang, Krishnamoorthy, Owens, and Sier surveyed over 700 papers of the personnel scheduling literature (Ernst et. al., 2004b). They developed the following table to illustrate the prior work in each application area. The application area is listed along with the number of papers surveyed in that particular application area.

| Application | Papers | Application | Papers |
|---------------------|--------|------------------------------|--------|
| Buses | 129 | Civic Services and Utilities | 22 |
| Nurse Scheduling | 103 | Venue Management | 19 |
| Airlines | 99 | Protection/Emergency Systems | 16 |
| Railways | 37 | Other Applications | 14 |
| Call Centers | 37 | Transportation Systems | 12 |
| General | 33 | Hospitality and Tourism | 7 |
| Manufacturing | 29 | Financial Services | 6 |
| Mass Transit | 28 | Sales | 3 |
| Health Care Systems | 23 | | |

Table 2.1: Number of surveyed papers in each application area of personnel scheduling (Ernst et. al., 2004b)

Ernst, et. al. (2004b) define each application area in their survey paper. I refer the reader to this article for additional detail of the categorization. The articles from each application area may concentrate on all or part of the problem instances described above as it pertains to the specific area.

The application areas pertinent to the research in this dissertation are characterized by flexible demand, a heterogeneous or mixed workforce, possible continuous operations, and personnel preference consideration. Alfares (2004) defines a mixed workforce as employees having differing “skill level, learning rate, wage, availability, and work hours.” Continuous operations are characterized by demand requirements 24 hours a day. Of the application areas defined by Ernst, et. al. (2004b), Hospitality and Tourism, Financial Services, and Sales are of particular relevance here. These areas represent a small portion of the personnel scheduling literature. They encompass only 16 of the over 700

papers surveyed. Furthermore, the subset of these papers examining the tour-scheduling problem is a further reduction of the papers surveyed.

2.3.1 Hospitality and Tourism Tour-scheduling

Hospitality and tourism industries as defined include “hotels, tourist resorts and restaurants” (Ernst et. al., 2004b). These industries have flexible demands and night and weekend shift requirements as well as employees with varying skill sets. Eveborn and Ronnqvist (2004) propose an elastic set-partitioning model and a branch and price algorithm to solve the tour-scheduling problem. The methodology is imbedded in a general scheduling software package called SCHEDULER. Glover and McMillan (1986) employ a tabu search and Thompson (1996) makes use of simulated annealing. Loucks and Jacobs (1991) begin by using a goal programming approach and then solve the corresponding integer program using a two-phase heuristic. A heuristic approach combined with a tabu search is developed by Litchfield, Ingolfsson, and Cheng (2003) to roster a restaurant. Finally, Love and Hoey (1990) define a mixed-integer program and solve the tour-scheduling problem using a minimum cost network flow simplex algorithm. This paper is of particular interest and will be discussed in further detail.

2.3.2 Financial Services Tour-scheduling

Financial services tour-scheduling is applied to staffing “clerical workers in service industries such as banking and insurance.” Again, the industry is characterized by flexible demand and by full and part-time workers. Li, Robinson, and Mabert (1991) and Mabert and Raedels (1977) use heuristics to

roster workers, the former considering differing skill sets. Mabert and Watts (1982) propose simulation to solve a set-covering formulation and Mould (1996) developed a spreadsheet decision support system (DSS) to allow employers to explore different tour schedules.

2.3.3 Sales Tour-scheduling

Retail sales tour-scheduling has received the least amount of attention in the literature (Ernst et. al., 2004b). Abboud et. al. (1998) propose a global search technique that is a hybrid of genetic algorithms and simulated annealing. Glover, McMillan, and Grover (1985) develop a DSS that uses a heuristic and Haase (1999) shows a column-generation technique to solve an integer-programming formulation.

2.4 Solution Techniques and Methodologies

The most important aspect of this literature review is to detail the prior work in methodology used to solve personnel scheduling problems. Ernst, et. al. (2004b) show a similar break down of solution techniques and methodologies as was shown for various application areas. The following table shows the techniques and the number of papers using that methodology. Papers employing more than one solution technique will appear in multiple categories.

| Method | Papers | Method | Papers |
|----------------------------------|--------|-----------------------|--------|
| Branch-and-Bound | 14 | Lagrangian Relaxation | 32 |
| Branch-and-Cut | 9 | Linear Programming | 35 |
| Branch-and-Price | 30 | Matching | 36 |
| Column Generation | 48 | Mathematical | 27 |
| Constraint Logic Programming | 46 | Network Flow | 38 |
| Constructive Heuristic | 133 | Other Meta-Heuristics | 11 |
| Dynamic Programming | 17 | Other Methods | 35 |
| Enumeration | 13 | Queuing Theory | 32 |
| Evolution | 4 | Set Covering | 58 |
| Expert Systems | 15 | Set Partitioning | 72 |
| Genetic Algorithms | 28 | Simple Local Search | 39 |
| Goal Programming | 19 | Simulated Annealing | 20 |
| Integer Programming | 139 | Simulation | 31 |
| Iterated Randomized Construction | 5 | Tabu Search | 16 |

Table 2.2: Number of surveyed papers using each methodology for personnel scheduling (Ernst et. al., 2004b)

Table 2.2 shows the disparate number of solution techniques and methodologies used to solve personnel scheduling problems. The methodology of particular interest to the research in this dissertation is network flow modeling. The following sections will show the application of the network flow modeling literature to the personnel scheduling problem instances with concentration on the tour-scheduling problem.

2.4.1 Network Flow Tour-scheduling (Airline, Mass Transit, Nurse Applications)

Network flow modeling has been used extensively in the airline industry. Yan and Chang (2002) and Barnhart, et. al. (1994) solve shortest-path network problems in a column generation formulation to pair airline crews to flight schedules. An additional column generation approach using network models to

price new columns is proposed by Mason and Smith (1998). Mellouli (2001) uses a “state-expanded aggregated time-space network” to solve airline and rail crew scheduling. Nicoletti (1975) uses assignment sub-problems within a constructive heuristic and Tingley (1979) sequentially solves assignment and matching problems to roster airlines.

Public and mass transit systems are also well addressed in the tour-scheduling literature. Patrikalakis and Xerocostas (1992) develop an approach that uses a network flow problem with side constraints to schedule vehicles for the Athens Area Urban Transport Organization. Carraresi and Gallo (1984) combine a matching approach and Lagrangean relaxation for crew rostering in mass transit. Borndorfer et. al. (2001) apply a column generation technique using network flow sub-problems for duty scheduling in European public transit. Ball and Benoit-Thompson (1988) sequentially solve shortest path and matching problems within a Lagrangean relaxation based heuristic. Banihashemi and Haghani (1991) propose to solve mass transit crew scheduling using a multi-commodity flow model.

One of the most studied applications in personnel scheduling is nurse scheduling. Nurse tour-scheduling or rostering is a subset of the literature for this application. The literature on nurse tour-scheduling is characterized by a known demand for nurses and a given set of nurse shifts. The heterogeneous workforce is characterized by varying nurse grades or skill sets and continuous or 24-hour operations. Ernst, et. al. (2004) give a breakdown of the various algorithmic approaches to nurse tour-scheduling. These mathematical

approaches include simulation, heuristics, global search techniques, constraint programming, and mathematical programming. The scope of this literature survey will be limited to the mathematical programming approaches, the differences between these approaches, and the proposed model in this research which I term the network-based mathematical program (NBMP).

Jumard, et. al. (1998) propose a 0 - 1 mixed-integer linear program (MIP) master problem to solve the nurse tour-scheduling problem. They incorporate a heterogeneous workforce, based upon nursing specialties or skills, overlapping shifts, and nurse preferences. The master problem contains a set of feasible shifts for the planning horizon that meet all the constraints. The goal of the master problem is to match nurses with each shift pattern. Other constraints such as minimum number of nurses with skill level ℓ scheduled at any time and the ratio of less experienced to more experienced nurses, are contained in the master problem. A branch and bound approach is taken to solve the master problem and maximize the preferences of the nurses. After the initial linear programming (LP) relaxation, an auxiliary problem is then solved using a resource constrained shortest path. The auxiliary problem determines where to branch and which column to generate into the basis.

Similarly, Millar and Kiragu (1998) develop patterns of shifts and days-off, which they term stints, where “all possible work-stretches can be pre-specified”. They attempt to find a set of stints that cover the demand for nurses using a mixed-integer branch and bound strategy. Warner (1976) also proposes an

approach where all feasible patterns of shifts and days off were enumerated and then allocated to a nurse.

One additional application of the day/shift and day-off pattern is found in “Solving the Problem of Re-rostering Nurse Schedules with Hard Constraints: New MultiCommodity Flow Models” by Margarida Moz and Margarida Vaz Pato. Published in the *Annals of Operations Research* in April 2004, the article proposes a method to re-roster a tour-schedule once changes in the given schedule no longer meet demand requirements. Sickness, termination, or hiring may be reasons for the need to change a given roster once it has been published.

Moz and Pato (2004) look at a continuous workforce that is heterogeneous. Differing nurse seniorities or specialties are taken into account by giving each shift in their model a separate task designation. The continuous workday is separated into 3 distinct shifts that start and end at the same time each day. These shifts are labeled Morning, Evening, and Night. A feasible schedule will meet hard constraints such as one shift per day, elimination of non-consecutive shifts, each shift/task combination assigned to a single nurse, the nurse assigned to a task must be eligible to perform the task, and a minimum number of days off per week for each nurse.

The algorithm approach proposed by Moz and Pato is an “integer multicommodity flow problem in a directed multi-level acyclic network” (2004). Each nurse is a commodity and is represented by an origin node in the network and by a destination node. The number of days to be scheduled is the number of

levels between the origin nodes and the destination node. Each non-origin/non-destination node in the network represents a task for a nurse to perform, including being off that day. The goal is to find a feasible path for each nurse through the network that meets all the hard constraints mentioned above. Below is a figure illustrating the network.

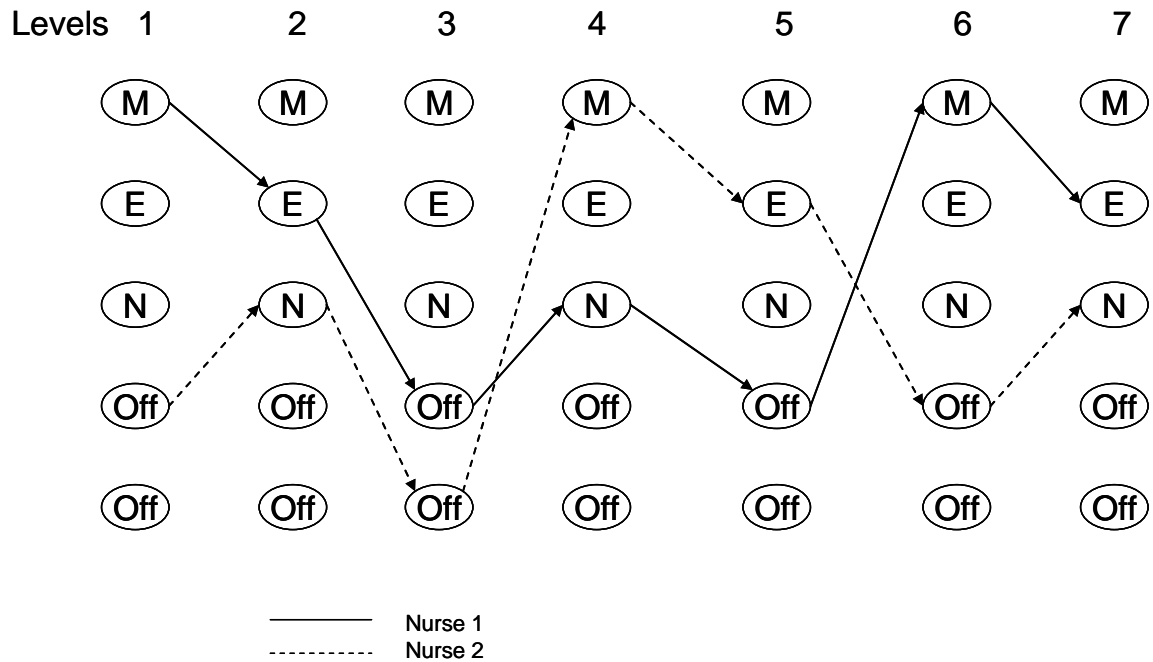


Figure 2.1: Multicommodity flow problem in a directed multi-level acyclic network (Moz & Pato, 2004)

Moz and Pato use this network formulation to re-roster a completed 28 day schedule that has been perturbed at some day d , which is considered the origin of the network. “The re-rostering problem amounts to finding an optimal integer flow of n commodities, from the nodes of level $d-1$ to those of the last level” (Moz & Pato, 2004). The formulation is a mixed integer linear program with binary constraints on the decision variables; they term this formulation as a

binary linear program (BLP). The objective function of the algorithm is to find a new schedule that will minimize the number of nurses who must swap shifts.

The BLP does not use the network structure to solve the problem. The network structure is for illustrative purposes, as noted by Moz and Pato “this flow approach provides a suggestive network model for RSP (re-rostering problem), it leads to large scale BLP formulation, with many variables and constraints, some of which are additional to the flow model” (2004). Due to the low number of shifts, the formulation in effect enumerates all possible paths through a network that will satisfy the constraints described above. The nodes of the network represent shift/skill combinations and the arcs represent links between shifts that are feasible. Moz and Pato acknowledge this. “It is worth noting that... the flow of each commodity can only use one path from its origin to its destination, thus resulting in binary flows” (Moz & Pato, 2004). The commodities are the various nurses working a specific shift performing a given task. The origin is day d , while the destination is the 28th day.

The Moz and Pato models, as well as Jumard, et. al. (1998), Millar and Kiragu (1998), and Warner (1976), vary significantly from the research proposed by the network-based mathematical program formulation; although, both models use the network formulation for mostly illustrative purposes. The prime difference is the models presented by Moz and Pato are mixed integer linear programs with binary constraints. This requires a branch and bound approach to solve the problem. In many instances the network-based mathematical program (NBMP) requires no branching because the solution to the linear program

provides binary decision variables. The proposed NBMP research can accommodate a large number of shifts with varied start times and of various lengths as well. The shifts in the NBMP research can also overlap. Moz and Pato describe a model where the shifts start and end at the same times each day and do not overlap. It is not readily apparent that their formulation could account for overlapping or varied length shifts. The enumeration of each feasible shift to shift to day off path through the network would make the problem grow rapidly if such variations in shift pattern were present. Next, although the workforce of nurses are heterogeneous in the aspect that they have varying skill sets, they do not have constraints on their availability apart from restrictions on consecutive shifts. The network-based mathematical program accounts for employee restrictions on availability as well as varying skill sets. Finally, the objective function of the Moz and Pato model is a deviation from a previously published tour-schedule. The objective of the proposed NBMP research is employee's weighted preference.

Although the use of network modeling has been well explored in the literature for airline, mass transit, and nurse tour-scheduling, these problems differ in nature from the problems of interest in this dissertation. In general, airlines, mass transit, and nursing applications are not characterized by flexible demand. They also employ a homogeneous workforce where each employee has similar availabilities, skill-sets, and work hours. These differences dramatically affect the formulation of the tour-scheduling problem. Additional constraints needed to model employee availability, minimum and maximum work

hours, consecutive shift restrictions, and skill-set eligibility are examples of the dissimilarities.

2.4.2 Network Flow Tour-scheduling with a Heterogeneous Workforce

Tour-scheduling a work force with varying daily and weekly availabilities, skill-sets, minimum and maximum work hours, and wage rates has not been given a proportionate amount of attention in the personnel scheduling literature. Furthermore, only Love and Hoey (1990) attack this problem from a network flow approach. Love and Hoey define a mixed-integer linear program (MILP) for shift scheduling and tour-scheduling for a fast food restaurant. The MILP is consistent with other formulations in the literature of the heterogeneous workforce problem and is shown below.

| | |
|------------|---|
| b_{hl} | = number of employees needed to work in time period h at work area l ; |
| x_{jkl} | = number of employees to work shift j on day k in work area l ; |
| a_{hjk} | = 1 if time period h is part of shift j on day k , 0 otherwise; |
| s_{hl} | = surplus employees for time period h in work area l ; |
| w_{ijkl} | = 1 if employee i is to work shift j on day k in work area l , 0 otherwise; |
| r_{jkl} | = shortage of employees to work shift j on day k in work area l ; |
| y_{ik} | = 1 if employee i is scheduled to work on day k , 0 otherwise; |
| z_i | = number of days employee i is scheduled to work; and |
| v_k | = 1 if employee i is to work at least k days in the week, 0 otherwise. |

minimize:

$$\sum_{hl} f_{hl} s_{hl} + \sum_{ijkl} c_{ijkl} w_{ijkl} + \sum_{jkl} g_{jkl} r_{jkl} + \sum_{ik} d_{ik} v_{ik} \quad (2.1)$$

subject to:

$$\sum_{jk} a_{hjk} x_{jkl} - s_{hl} = b_{hl} \quad \text{all } h, l \quad (2.2)$$

$$\sum_i w_{ijkl} + r_{jkl} - x_{jkl} = 0 \quad \text{all } j, k, l \quad (2.3)$$

$$\sum_{jl} w_{ijkl} - y_{ik} = 0 \quad \text{all } i, k \quad (2.4)$$

$$\sum_k y_{ik} - z_i = 0 \quad \text{all } i \quad (2.5)$$

$$\sum_k v_{ik} - z_i = 0 \quad \text{all } i \quad (2.6)$$

$$y_{ik} \leq 1 \quad \text{all } i, k \quad (2.7)$$

$$z_{ik} \leq 1 \quad \text{all } i, k \quad (2.8)$$

Love and Hoey then propose the decomposition of this MILP into two sub-problems. Minimizing

$$\sum_{hl} f_{hl} s_{hl} \quad (2.9)$$

subject to constraint 2.2 will solve the shift assignment problem. This first sub-problem will define the shifts needed to cover demand. The second sub-problem minimizes the rest of the objective function subject to equations 2.3 – 2.8. Love and Hoey (1990) state that since “each column in equations (2.3) – (2.6) has at most two nonzero entries and these equations can be constructed such that each column with the two nonzero entries has one +1 and one -1, this second sub-problem can also be solved as a minimum cost network flow problem.” The authors propose solving these sub-problems using network simplex algorithms.

Although Love and Hoey (1990) show that a tour-scheduling problem can be solved with a minimum cost network flow, there are limitations to the complexity of the heterogeneous problem that can be solved with this formulation. As acknowledged by the authors, this formulation does not perform well for continuous operations. Equation 2.4 insures that each employee works only one shift per day. However, this constraint does not preclude an employee that just worked an 8 hour evening shift from being scheduled for the consecutive late night shift which is technically on a different day. Additionally, there is no discussion of employees who have differing skill sets or the more complex case of employees who have overlapping and differing skill sets. An example would be employee 1 who can run the cash register and drive-thru, while employee 2 can run the cash register and the fry machine. A more robust tour schedule must take into account the specific skill-sets of the employees and the skills required per shift.

The research detailed in this dissertation uses an expanded network model to allow for varying skill sets and specialized side constraints to manage continuous operations. The expanded network formulation with specialized side constraints forms a linear program whose solutions are typically integer.

2.5 Linear Programming Approach to Tour-scheduling

Alfares (2004), in his survey paper, notes that most of the linear programming approaches in the literature are heuristic or are approximations. Many, such as Brusco and Johns (1995), Mabert and Showalter (1990), Ashley

(1995), and Lauer et. al. (1994) use LP-based heuristics to solve the tour-scheduling problem. Ashley (1995) in particular uses an iterative LP approach to solve a heterogeneous tour-scheduling problem for the library staff at the University of Kansas. The implementation of the methodology requires human interaction with a spreadsheet optimizer. In each iteration, the operator must manually set fractional variables to be integer until a feasible schedule is created. Although Ashley (1995) reports success in a relatively few number of iterations, with only 14 employees the problem size discussed is small. There is no guarantee of optimality for large problems.

Lauer et. al. (1994) describe the heterogeneous nature of scheduling university students to a computer lab. Students have restricted availability due to class participation requirements, differing skill-sets, and many labs are open and must be staffed 24 hours. Lauer et. al. (1994) allow each employee to choose their own schedule from a set of possible shifts which have been determined using an LP. This approach may optimize preferences based upon seniority or the sequence in which the employees chose available shifts, but will not guarantee a feasible solution. If the final employees to be scheduled are not available or do not possess the required skills for the remaining shifts, the schedule will have to be iterated. The research presented in this dissertation will show application by solving a similar university computer lab staffing problem to optimality.

2.6 Literature Summary

The amount of work previously done and currently being pursued to solve personnel scheduling problems is extensive. Obviously, the area of research continues to be important to both the academic and business communities. Although much work has been done, the concentration on tour-scheduling or crew rostering of a flexible demand industry with a heterogeneous workforce and continuous operations using network flow or linear programming approaches is much more limited. Table 2.3 shows the void in the literature this dissertation will address. The table uses the previously proposed categorization by Ernst et. al. (2004b) of the tour-scheduling papers reviewed in this chapter.

| Tour Scheduling/Rostering Problem Instance | Buses | Nurse Scheduling | Airlines | Railways | Call Centers | General | Manufacturing | Mass Transit | Health Care Systems | Civic Services and Utilities | Venue Management | Protection/Emergency Systems | Other Applications | Transportation Systems | Hospitality and Tourism | Financial Services | Sales |
|---|-------|------------------|----------|----------|--------------|---------|---------------|--------------|---------------------|------------------------------|------------------|------------------------------|--------------------|------------------------|-------------------------|--------------------|-------|
| Branch-and-Bound | | | | | | | | | | | | | | | | | |
| Branch-and-Cut | | | | | | | | | | | | | | | | | |
| Branch-and-Price | | | | | | | | | | | | | | | 1 | | |
| Column Generation | | | | | | | | | | | | | | | | | 1 |
| Constraint Logic Programming | | | | | | | | | | | | | | | | | |
| Constructive Heuristic | | | | | | | | | | | | | | | 1 | 2 | 1 |
| Dynamic Programming | | | | | | | | | | | | | | | | | |
| Enumeration | | | | | | | | | | | | | | | | | |
| Evolution | | | | | | | | | | | | | | | | | |
| Expert Systems | | | | | | | | | | | | | | | | | |
| Genetic Algorithms | | | | | | | | | | | | | | | | | 1 |
| Goal Programming | | | | | | | | | | | | | | | 1 | | |
| Integer Programming | | | | | | | | | | | | | | | | | |
| Iterated Randomized Construction | | | | | | | | | | | | | | | | | |
| Lagrangian Relaxation | | | | | | | | | | | | | | | | | |
| Linear Programming | | | | | | | | | | | | | | | | | |
| Matching | | | | | | | | | | | | | | | | | |
| Mathematical | | | | | | | | | | | | | | | | | |
| Network Flow | | 4 | 6 | | | | | 5 | | | | | | | 1 | | |
| Other Meta-Heuristics | | | | | | | | | | | | | | | | | |
| Other Methods | | | | | | | | | | | | | | | | | |
| Queuing Theory | | | | | | | | | | | | | | | | | |
| Set Covering | | | | | | | | | | | | | | | | | 1 |
| Set Partitioning | | | | | | | | | | | | | | | 1 | | |
| Simple Local Search | | | | | | | | | | | | | | | | | |
| Simulated Annealing | | | | | | | | | | | | | | | 1 | 1 | |
| Simulation | | | | | | | | | | | | | | | | 1 | |
| Tabu Search | | | | | | | | | | | | | | | 2 | | |

Table 2.3: Number of papers reviewed that specifically addressed the tour-scheduling problem instance using the methodology breakdown and application areas proposed by Ernst, et. al. (2004b)

Chapter 3

A NETWORK-FLOW BASED MATHEMATICAL PROGRAMMING APPROACH TO OPTIMAL ROSTERING OF CONTINUOUS HETEROGENEOUS WORKFORCES

Abstract

We present a linear programming (LP) approach to a generalized personnel tour scheduling integer programming (IP) problem based upon a minimum cost network-flow formulation with specialized side constraints. The formulation simultaneously incorporates many realistic constraint types and optimally solves the IP tour scheduling problem for industries with continuous (24 hour) operations and a heterogeneous workforce with varying availabilities, shift preferences, restrictions on working consecutive shifts, differing skill-sets, and minimum and maximum shift requirements per week. Our work is the first to solve this generalized class of IP problems optimally. Problems of this complexity routinely occur in industry but are computationally prohibitive to solve. In many instances integer solutions to our tour scheduling problem are found without branching, bounding, or cutting schemes. This allows the CPLEX interior-point method to generate solutions orders of magnitude faster than the corresponding integer program does. Our methodology is demonstrated by finding an optimal tour schedule for staffing the four main Computer Commons at Arizona State University, a problem of order 10,000 binary variables and 1,000 constraints in 0.12 CPU seconds. It is also used to perform the calculations for a generalized staff sizing problem.

3.1 Introduction

Personnel scheduling is one of the most difficult, important, and studied problems in operations research. The optimal choice of the number of employees required to meet customer demand, shift start and stop times, daily lunches and breaks, and the assignment of the adequately skilled employee to the best shift is a classic combinatorial optimization problem. Managers who attempt to manually solve the personnel scheduling problem expend many valuable work hours to find even a feasible solution which has little probability of being optimal based on any objective function. Sub-optimal scheduling increases an industry's tangible costs, not only through the consumption of a manager's time, but also through the misallocation of shifts to meet customer demand and through employees staffing shifts for which they are not qualified. Furthermore, a sub-optimal schedule will increase intangible costs such as lower employee morale from dissatisfaction with a poor work schedule. Eliminating these costs is an easy way to increase the profitability of any business.

Personnel scheduling is a necessary chore associated with any organization employing people to get work accomplished. Some organizations that employ a professional workforce during traditionally scheduled business hours find the personnel scheduling problem a trivial concern. However, industries that rely on a large portion of part-time employees with highly constrained availabilities (work hours) and require staffing for the entire day and evening hours or, in particular, continuous (24-hour) operations have a much more challenging problem. Such industries include fast-food restaurants, hotels

and resorts, grocery stores, customer assistance telephone services, hospitals, and others. The combined salary cost of this segment of the economy is quite large. A timely and optimal solution to their personnel scheduling problem will save managers valuable time, meet customer demand at minimum labor cost, and increase worker morale by matching each employee to the best shift possible. This article presents an extremely fast, computationally efficient, and optimal network-flow based linear programming solution for the extremely challenging integer programming problem of rostering a continuous heterogeneous workforce with realistic constraints.

Personnel scheduling is generally decomposed into more tractable sub-problems. The three primary sub-problems are demand modeling, shift selection, and employee tour scheduling or rostering. Demand modeling determines the number of employees required on duty during a given time interval to satisfy customer needs. Forecasting and queuing theory are the primary tools applied to demand modeling. A typical chart of the number of employees required, illustrating the results of demand modeling, is shown in Figure 3.1.

The second primary personnel scheduling sub-problem is shift selection. Shift selection assigns consecutive hour shifts to satisfy the customer demand while meeting organizational and regulatory requirements for shift lengths, breaks, and mealtime allowances. The objective of shift selection is generally to minimize the number of hours scheduled that exceed customer demand. Figure

3.2 shows an example of a set of shifts that will optimally cover the number of employees required in Figure 3.1.

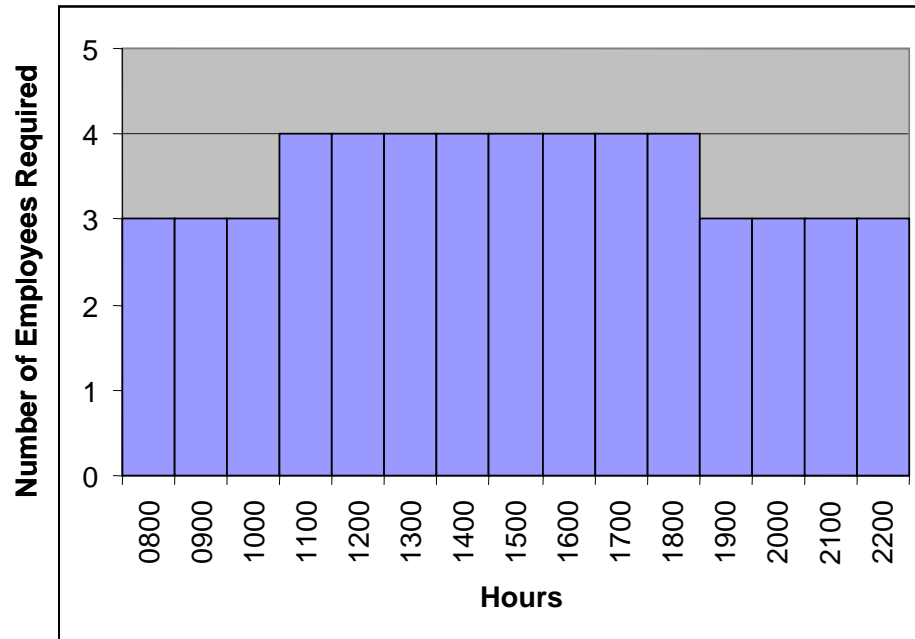


Figure 3.1: Employees requirement by hour from demand modeling

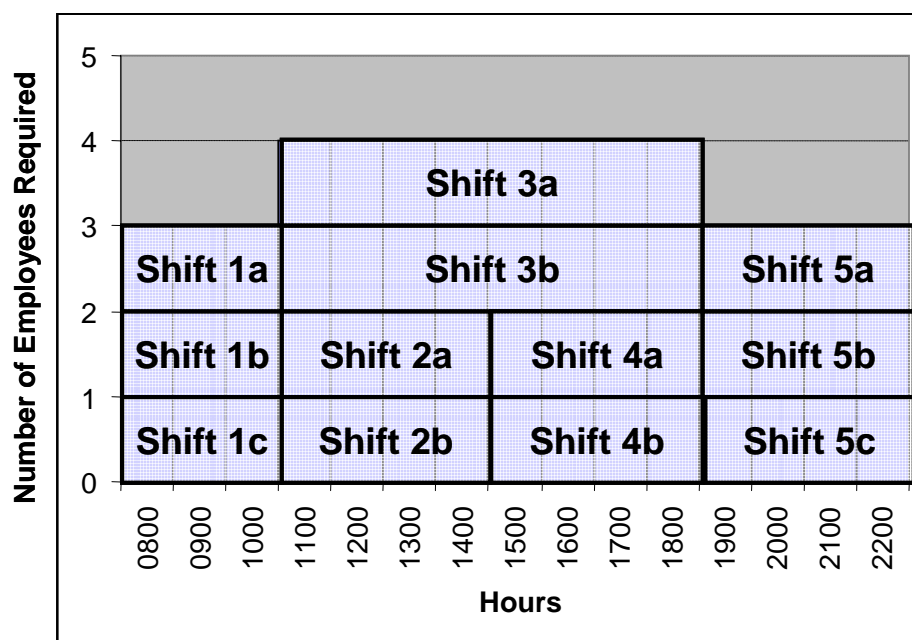


Figure 3.2: Example set of shifts that cover the demand in Fig. 3.1

The third personnel scheduling sub-problem is tour scheduling also known as employee rostering. Tour scheduling assigns individuals to specific shifts detailed during shift selection. A heterogeneous workforce is a collection of personnel who have significantly different availabilities, skill sets, and seniorities. Tour scheduling a heterogeneous workforce must consider all of these differences and optimally match the personnel with the best shift possible for which they are eligible. Additionally, considerations for individuals working a minimum and maximum number of shifts per week and insuring adequate rest between shifts for industries with continuous (24-hour) operations are critical to an optimal tour schedule. Heterogeneous and continuous workforces are realistic considerations for many industries, as previously noted. Employee shift

preferences and management employee weighting comprise the objective function.

This methodology provides several original contributions to the personnel scheduling literature. The following list summarizes the advancements proposed by this research.

- Provide a solution methodology that simultaneously accounts for variations in employee availabilities, skill sets, and preferences.
- Provide a methodology that rosters employees for continuous operations by allowing for rest hours between shifts.
- Provide a network-flow based formulation that rapidly solves the tour scheduling problem, many times providing integer answers without any need for branching, bounding, or cutting schemes.

This article is organized in the following manner. Section 2 contains a survey of the pertinent literature and important previous work accomplished in personnel tour scheduling. Section 3 details the tour scheduling problem at Arizona State University for computer lab technicians. Our methodology is general, but we feel that for realism and clarity we will introduce our case study early. Section 4 provides the proposed solution method and model. Section 5 shows the results of the methodology as applied to the case study as well as applied to larger instances of the IP tour scheduling problem. Section 6 provides conclusions and recommendations.

3.2 Literature Survey

A vast amount of scholastic work has been accomplished on the personnel scheduling problem. This work has been documented in a wide array of refereed journals, conference proceedings, and lecture notes. Personnel scheduling spans a wide range of problem instances, specific applications, and solution techniques and methodologies. Many survey papers of the pertinent literature in the area have been written. Some of the most extensive and most recent include Ernst, et. al. (2004a), Ernst, et. al. (2004b), and Alfares (2004).

3.2.1 Application Areas

Personnel scheduling has a broad range of application. In their article, Ernst, et. al. (2004b) survey over 700 papers in the personnel scheduling literature. They define application areas and catalog prior work in each application area. We refer the reader to this article for additional detail of the categorization. The articles from each application area may concentrate on all or part of the problem instances described as it pertains to the specific area.

The application areas pertinent to our research are characterized by flexible demand, a heterogeneous or mixed workforce, possible continuous operations, and personnel preference consideration. Alfares (2004) defines a mixed workforce as employees having differing “skill level, learning rate, wage, availability, and work hours.” Continuous operations are characterized by demand requirements 24 hours a day. Important application areas of interest as defined by Ernst, et. al. (2004b) are Hospitality and Tourism, Financial Services, and Sales. Although these areas represent a significant proportion of economic

activity, they form only a small portion of the personnel scheduling literature, only 16 of the over 700 papers surveyed. Furthermore, the subset of these papers examining the tour scheduling problem is a further reduction of the papers surveyed.

3.2.1.1 Hospitality and Tourism Tour Scheduling

Hospitality and tourism industries as defined include “hotels, tourist resorts and restaurants” (Ernst et. al., 2004b). These industries have flexible demands and night and weekend shift requirements as well as employees with varying skill sets. Eveborn and Ronnqvist (2004) propose an elastic set-partitioning model and a branch-and price algorithm to solve the tour scheduling problem. The methodology is imbedded in a general scheduling software package called SCHEDULER. Glover and McMillan (1986) employ a tabu search and Thompson (1996) makes use of simulated annealing. Loucks and Jacobs (1991) begin by using a goal programming approach and then solve the corresponding integer program using a two-phase heuristic. A heuristic approach combined with a tabu search is developed by Litchfield, Ingolfsson, and Cheng (2003) to roster a restaurant. Finally, Love and Hoey (1990) define a mixed-integer program and solve the tour scheduling problem using a minimum cost network flow simplex algorithm. This paper is of particular interest and will be discussed in further detail.

3.2.1.2 Financial Services Tour Scheduling

Financial services tour scheduling is applied to staffing “clerical workers in service industries such as banking and insurance.” Again, the industry is

characterized by flexible demand and by full and part-time workers. Li, Robinson, and Mabert (1991) and Mabert and Raedels (1977) use heuristics to roster workers, the former considering differing skill sets. Mabert and Watts (1982) propose simulation to solve a set-covering formulation and Mould (1996) develops a spreadsheet decision support system (DSS) to allow employers to explore different tour schedules.

3.2.1.3 Sales Tour Scheduling

Retail sales tour scheduling has received the least amount of attention in the literature. Glover, McMillan, and Grover (1985) develop a DSS that uses a heuristic and Haase (1999) shows a column-generation technique to solve an integer-programming formulation.

3.2.2 Solution Techniques and Methodologies

The most important aspect of this literature review is to detail the prior work in methodology to solve personnel scheduling problems. Ernst, et. al. (2004b) show a break down of solution techniques and methodologies as was shown for various application areas. Papers employing more than one solution technique appear in multiple categories. The methodology of particular interest to our research in this article is network flow modeling. The following sections will show the application of the network flow modeling literature to the personnel scheduling problem instances with a concentration on the tour scheduling problem.

3.2.2.1 Network Flow Tour Scheduling (Airline, Mass Transit, Nurse Applications)

Network flow models have been used extensively in the airline industry. Yan and Chang (2002) and Barnhart, et. al. (1994) solve shortest-path network problems in a column generation formulation to pair airline crews to flight schedules. Column generation using network models to price new columns is proposed by Mason and Smith (1998). Mellouli (2001) uses a “state-expanded aggregated time-space network” to solve airline and rail crew scheduling. Nicoletti (1975) uses assignment sub-problems within a constructive heuristic, and Tingley (1979) sequentially solves assignment and matching problems to roster airlines.

Public and mass transit systems are also well addressed in the tour scheduling literature. Patrikalakis and Xerocostas (1992) develop an approach that uses a network flow problem with side constraints to schedule vehicles for the Athens Area Urban Transport Organization. Carraresi and Gallo (1984) combine a matching approach and Lagrangean relaxation for crew rostering in mass transit. Borndorfer et. al. (2001) apply a column generation technique using network flow sub-problems for duty scheduling in European public transit. Ball and Benoit-Thompson (1988) sequentially solve shortest path and matching problems within a Lagrangean relaxation based heuristic. Banihashemi and Haghani (1991) propose to solve mass transit crew scheduling using a multi-commodity flow model.

Another popular application area for network flow modeling is nurse scheduling. Dowsland and Thompson (2000) mix knapsack, network flow, and

tabu search techniques to develop nurse rosters. Jaumard, Semet, and Vovor (1998) solve shortest-path network sub-problems within a column generation integer program. Millar and Kiragu (1998) develop cyclic and acyclic work schedules for nurses using a shortest-path problem with side constraints. Finally, Moz and Pato (2004) solve a nurse re-rostering problem using multi-commodity flow models.

Although the use of network modeling has been well explored in the literature for airline, mass transit, and nurse tour scheduling, these problems differ in nature from our problems of interest. In general, airlines, mass transit, and nursing applications are not characterized by flexible demand and additionally employ a homogeneous workforce where each employee has similar availabilities, skill-sets, and work hours. These differences dramatically affect the formulation of the tour scheduling problem. Additional constraints needed to model employee availability, minimum and maximum work hours, consecutive shift restrictions, and skill-set eligibility are examples of the dissimilarities.

3.2.2.2 Network Flow Tour Scheduling with a Heterogeneous Workforce

Tour scheduling a work force with varying daily and weekly availabilities, skill-sets, minimum and maximum work hours, and wage rates has not been given a proportionate amount of attention in the personnel scheduling literature. Furthermore, only Love and Hoey (1990) attack this problem from a network flow approach. Love and Hoey define a mixed-integer linear program (MILP) for shift scheduling and tour scheduling for a fast-food restaurant. They then decompose the MILP into two sub-problems. This is the same approach that we will take.

The first sub-problem defines the shifts needed to cover demand. The second sub-problem minimizes the rest of the objective function. As Love and Hoey (1990) state: “each column has at most two nonzero entries and these equations can be constructed such that each column with the two nonzero entries has one +1 and one -1, so this second sub-problem can be solved as a minimum cost network flow problem.” They propose solving these sub-problems using network simplex algorithms.

Although Love and Hoey (1990) show that a tour scheduling problem can be solved with a minimum cost network flow, there are limitations to the complexity of the heterogeneous problem that can be solved with their formulation. As acknowledged by the authors, this formulation does not perform well for continuous operations. Additionally, there is no discussion of employees who have differing skill sets or the more complex case of employees with overlapping and differing skill sets. A more robust tour schedule must take into account the specific skill-sets of the employees and the skills required per shift.

Our methodology uses an expanded network model to allow for varying skill sets and specialized side constraints to manage continuous operations. As we will see, the expanded network formulation with specialized side constraints forms a linear program (LP) whose solutions are in many cases integer.

3.3 A Case Study - ASU Computer Lab Technician Tour Scheduling

There are four computer labs at Arizona State University (ASU): the Atrium, BAC, GWC, and ECG. Each lab is staffed by technical experts (typically

enrolled ASU science and engineering students) who assist other students using the computers in the lab. Currently, the Technology Support Analyst Principal (TSAP) is tasked with scheduling the technicians to meet the historic demand for each lab based upon the time of day. Each technician has a subset of the four labs in which they are qualified to work. The TSAP currently schedules over fifty technicians to the various labs by hand. He must take into consideration the demand for technicians as a function of day and time, availability and non-availability of the technicians, shift restrictions and maximum hours per week, minimum hours per week, as well as the preferences and qualifications of the technicians for particular shifts. Each such effort requires 2-3 days and a satisfactory solution is the first feasible one found.

Each lab has its own operating hours and staffing requirements. For example, Table 3.1 depicts the operating hours and staff requirements for the Atrium. Although there must be no staff during closed hours, and at least one technician working each operating hour, there is some flexibility regarding the staffing levels during operating hours. Staffing the lab in excess of the requirements in the table will result in excess costs to the university. Staffing the labs at a lower level is not allowed.

| | 7a | 8a | 9a | 10a | 11a | 12p | 1p | 2p | 3p | 4p | 5p | 6p | 7p | 8p | 9p | 10p | 11p |
|-----------|----|----|----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----|-----|
| Sunday | | | 3 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 3 | 3 | 4 | 4 | 4 | 4 | 3 |
| Monday | 3 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 3 | 3 | 4 | 4 | 4 | 4 | 3 |
| Tuesday | 3 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 3 | 3 | 4 | 4 | 4 | 4 | 3 |
| Wednesday | 3 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 3 | 3 | 4 | 4 | 4 | 4 | 3 |
| Thursday | 3 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 3 | 3 | 4 | 4 | 4 | 4 | 3 |
| Friday | 3 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | 4 | 4 | 3 | 3 | 4 | | | | |
| Saturday | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | | |

Table 3.1: Lab technician requirements (Atrium) based on hourly demand

Each technician has his/her own availability profile and preferences among those hours. The minimum and maximum shifts per week are developed by the TASP after his consultation with each technician.

A two-phase approach to the tour scheduling problem for ASU computer labs is proposed as depicted in Figure 3.3. In Phase 1, we select a set of shifts to cover the staffing requirement for each of the four labs. The scheduler can indicate shift length priorities in order to bias the model in favor of specific shift lengths. For example the scheduler can specify a preference for 6-hour shifts as opposed to 3-hour shifts by assigning them different weights. Additionally, benefits arise from the fact that selecting the shifts for a single lab has no effect on which shifts should be selected for the other three. This independence allows for the selection of shifts for each of the four labs to be done individually, resulting in four smaller Phase 1 models.

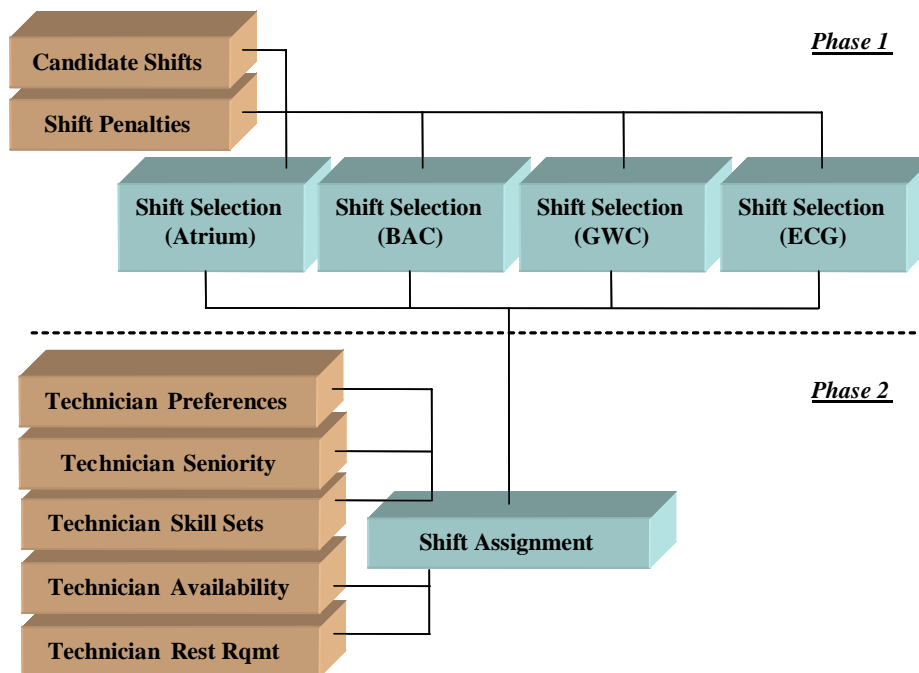


Figure 3.3: Two-phase methodology for shift selection and rostering

In Phase 1, we consider all reasonable shift lengths over the planning horizon. The objective of this phase is to select a set of shifts that covers the lab requirements, as well as minimizes the sum of “shift penalties”. Shift penalties are assigned based on the scheduler’s preference for a particular shift length. A penalty of 1.0 signifies that no extra weight is given to the hours in a given shift length. Phase 1 then minimizes the total cost of selecting individual shifts, where the cost of each shift is the product of the number of hours and the scheduler’s penalty factor for the shift. The penalties for shift lengths solicited from the TSAP for our case study are contained in Table 3.2.

| Shift Length | Penalty Factor | Cost |
|--------------|----------------|------|
| 3 | 1.05 | 3.15 |
| 4 | 1.00 | 4.00 |
| 5 | 1.00 | 5.00 |
| 6 | 1.00 | 6.00 |
| 7 | 1.10 | 7.70 |
| 8 | 1.15 | 9.20 |

Table 3.2: Shift penalties used in Phase 1 set covering problem

For example, consider the two possible shift selections with regard to Saturday mid-day lab requirements in Figure 3.4 below. The solutions show the shift lengths chosen as well as the number of employees on each shift. The value of solution 1 is 53.45, whereas the value of solution 2 is 59.9. Solution 1 happens to be the optimal solution. Both solutions cover the demand profile exactly, but Solution 1 uses shift lengths preferred by the manager.

| | | | | | | | | | | | | | | | | | |
|------------|-----------------|---|---|-----------------|---|---|---|-----------------|---|---|---|-----------------|---|---|---|--|--|
| Saturday | | | | | | | | | | | | | | | | | |
| Employee | | | | | | | | | | | | | | | | | |
| Demands | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 | | |
| solution 1 | shift 1 (3 emp) | | | shift 2 (4 emp) | | | | shift 3 (4 emp) | | | | shift 4 (3 emp) | | | | | |
| solution 2 | shift 1 (3 emp) | | | | | | | shift 2 (3 emp) | | | | | | | | | |
| | | | | shift 3 (1 emp) | | | | | | | | | | | | | |

Figure 3.4: Example of effect of preference on two distinct shift definitions

Though NP-complete, the Phase 1 problem is not computationally challenging. A set-covering integer program (IP) is directly used to solve Phase 1. It selects the optimal set of shifts to cover customer demand based on shift length preferences. In Phase 2, we must assign employees to these shifts, subject to employee preferences by hour, availabilities, skills, rest periods as well as some notion of employee preference by management. This is a very difficult problem to solve efficiently and is the focus of this article.

3.4 Proposed Tour-Schedule Solution Method and Model

The tour-scheduling problem is inherently a binary set-covering problem. Therefore, as problem size increases, the efficiency of the associated IP decreases rapidly. However, we have developed a formulation of the problem as a minimum cost network-flow, using an arc capacity method, so that the resulting network structure many times provides integer binary answers. The network structure can easily be written as an LP and solved using fast solution algorithms such as the CPLEX interior point method. Solutions of this formulation do not require any branching, bounding, or cutting schemes to find integer solutions – the frequency of the integer solutions are a consequence of the formulation method. Therefore, very large problem instances can be solved in a computationally insignificant amount of time compared to the corresponding IP solution. Figure 3.5 shows our formulation of our generalized tour-scheduling problem as a minimum cost network-flow.

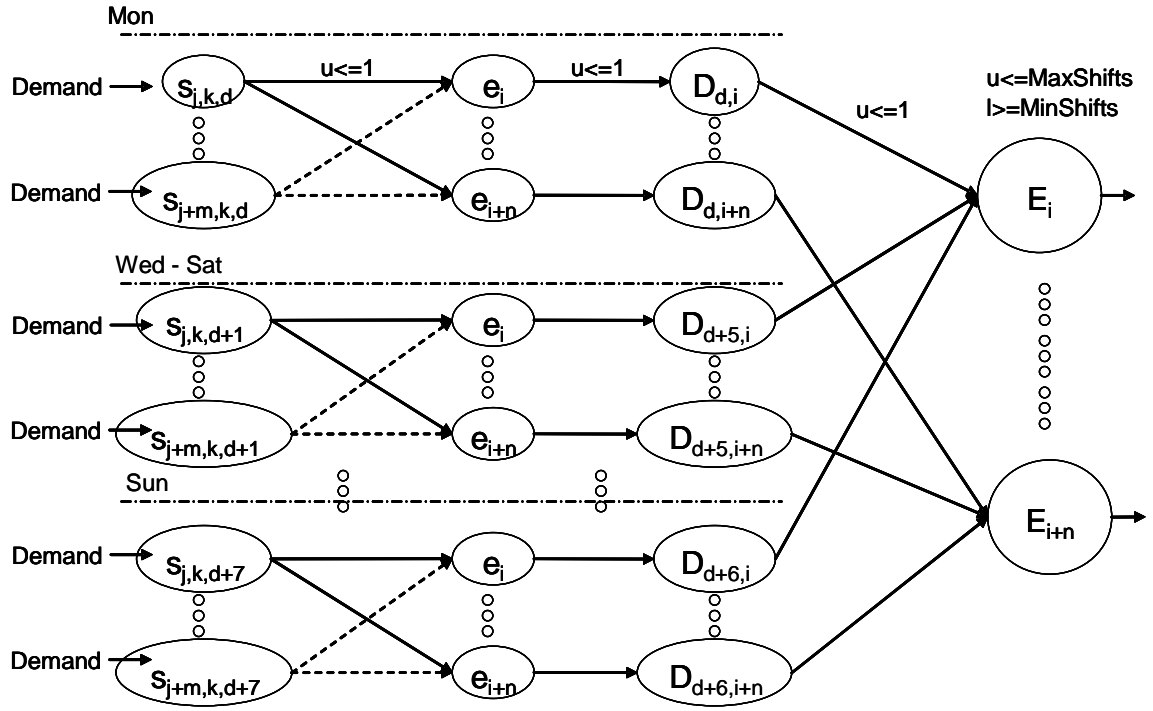


Figure 3.5: Network representation of the Phase 2 tour-scheduling problem

Here, $s_{j,k,d}$ denotes shift number j requiring skill set k and on day d , e_i is employee number i , $D_{d,i}$ is the total number of shifts, D , on day d for employee i , and E_i is the total number of shifts per week, E , for employee i .

The number of employees needed for each shift node flows from the Demand arc. Each shift node has an edge to each qualified and available employee node. The shift-to-employee arc is capacitated at 1, meaning only one of the required staffing for a shift can be assigned to a single employee. Each employee then flows their daily work assignment to the employees' daily-total-node, D . This arc is capacitated at 1, meaning each employee can work only one shift per day. Finally, the employees' daily-total-nodes channel the flow to

the employees' weekly-total-node which contains the capacities to enforce min and max shifts per week.

The balance equations for the above network flow formulation are:

$$\sum_i s_{j,k,d} e_i = demand \quad \text{for all } j,k,d \quad (3.1)$$

$$\sum_{j,k} s_{j,k,d} e_i = D_{d,i} \quad \text{for all } i,d \quad (3.2)$$

$$\sum_d D_{d,i} \leq \max number_of_shifts \quad \text{for all } j \quad (3.3)$$

$$\sum_d D_{d,i} \geq \min number_of_shifts \quad \text{for all } j \quad (3.4)$$

$$D_{d,i} \leq 1 \quad \text{for all } i,d \quad (3.5)$$

Constraint set 1 requires that the demand for a number of employees for each type of shift is met. Set 2 totals the shifts per employee for each day and requires that the total be less than or equal to 1 in constraint set 5 (no employee works more than one shift per day, but remember that the shift length is a variable). Constraint sets 3 and 4 require that the minimum and maximum numbers of shifts per week are met for each employee. The following three sub-sections describe how our model incorporates all the constraints needed to create a tour-schedule for a real world business, including a heterogeneous work force and incorporate consecutive shift restrictions.

3.4.1 Consecutive Shift Restrictions

Adequate rest between consecutive shifts is a requirement for any realistic tour-schedule of continuous operations. Industries that are only open for an 8-hour day have tour-schedules that implicitly contain adequate rest. However, when scheduling an industry with continuous or 24-hour operations, restrictions on consecutive shifts is a requirement for any realistic tour-scheduling

formulation. Love and Hoey (1990) and Loucks and Jacobs (1991) note the need for consecutive shift restrictions and state the lack of such restrictions as limitations to their tour-scheduling models. Our formulation allows for specialized side constraints to accommodate continuous operations.

As input in any specific application, we use the appropriate industry standard for how much time is required between the start of shift j and the start of the next shift. This leads to a matrix of shifts, J , that conflict with each other in the sense that an employee assigned to work shift j cannot be scheduled to work any shift conflicted with shift j . The following constraint is added to constraints 1-5 above to accommodate consecutive shift restrictions.

$$\sum_{j \in J} s_{j,k,d} e_i \leq 1 \quad \text{for all conflicted shifts} \quad (3.6)$$

Surprisingly, our empirical experience indicates that these side constraints do not destroy the integrality of the solution to the minimum cost network flow in many cases. However, in theory non-integer answers are possible.

3.4.2 Employee Availabilities and Varying Skill Sets

Many industries have employees with heterogeneous availabilities. For example, service industries often employ younger, part-time employees who have restrictions on their time due to schooling or other activities. Our model can be modified to accommodate such heterogeneous availabilities. We simply remove the network arc from a shift to an employee who is not available during the hours encompassing that shift. No flow units can travel from that shift to the employee and consequently the employee is not assigned to that shift.

Similarly, it is often the case that many service industries have different skill requirements needed during a particular shift. If an employee does not possess the required skill for a particular shift, the arc from the shift to the employee is removed from the network. If the employee has multiple skill sets, the arc from the employee to all shifts within their set is contained within the network.

This is an important aspect to the efficiency of the tour-schedule. Many examples within the tour-scheduling literature schedule only one department within the industry at a time. Therefore, the employees assigned to that department can be scheduled only within that department and there is no mechanism to schedule that employee across departments. This could lead a department to over-staffing costing a business extra payroll. We have no such restriction.

Removal of the two types of arcs described above are easy to program. They have no effect on the integrality of solutions from the LP-network solution. This is because the resulting network is still of minimum cost flow form.

3.4.3 The Objective Function - Employee and Manager Preferences and a Perturbation

The objective function of the constraint set described above can now be defined. First, each employee scores each shift, on a preference scale, based upon their desire to work at that time. Any scale could be used, but we have found in practice that one based on a scale of (0,100) allows for adequate granularity. The preliminary objective function is then:

$$\text{MAX} \quad \sum_{j,k,d,i} s_{j,k,d} e_i \bullet \text{pref}(s_{j,k,d} e_i) \quad (3.7)$$

Additionally, we allow the manager to express their preferences of employees on shifts taking such factors as seniority or job performance into account. We call these preferences “rewards”. The larger the reward on a scale of (0,100), the more likely the model will select that employee for desired shifts (and it certainly helps to break ties). The objective function becomes:

$$\text{MAX} \quad \sum_{j,k,d,i} s_{j,k,d} e_i \bullet [\text{pref}(s_{j,k,d} e_i) + \text{Re ward}(s_{j,k,d}, e_i)] \quad (3.8)$$

Finally, it is possible for the algorithm to find an alternate optimal solution that contains fractional employee assignments. This occurs when the preferences of two employees for a particular shift are equal and neither is violating a consecutive shift restriction or a maximum number of shifts per week constraint. To remove this non-integer alternate optimal solution effect from the view of manager we add to the objective function a small random value, ε , whose magnitude is chosen to range from (0,1) based on the magnitude of the preference and reward scales of (0,100). The random value is not reported as part of the final objective function value, merely used to remove the possibility of non-integer alternative optimal solutions

$$\text{MAX} \quad \sum_{j,k,d,i} s_{j,k,d} e_i \bullet [\text{pref}(s_{j,k,d} e_i) + \text{Re ward}(e_i)] + \varepsilon \quad (3.9)$$

Data organization to support these calculations are straightforward. A spreadsheet is used as the input data file. A Visual Basic macro then writes the linear program in CPLEX .lp input format. This macro runs almost instantaneously. The output file is then given to CPLEX for optimization. The ease of changing data in the input data file, the speed with which the output file is written, and the speed with which the problem is optimized (more on this in the next section) allows the user to study a wide variety of problem instances and conduct sensitivity analysis. Figure 3.6 shows an example of the input data file format. The data file for the ASU computer lab problem has 195 rows of shifts with 50 columns of preferences (the sum of employee and manager values).

| | | | | | | Emp Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----|----------|--------------|---------|--------------|------------|---------------|----------|------|------|------|------|------|------|------|------|------|
| | | | | | | Emp Skill Set | All | All | All | All | All | All | All | All | All | All |
| | | | | | | Reward | High | High | High | High | High | High | High | High | High | High |
| Day | Shift ID | Shift Number | Com Lab | Shift Demand | Start Time | Employee | Priority | | | | | | | | | |
| 1 | 56 | 1 | Atr | 1 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 57 | 2 | Atr | 1 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 58 | 3 | Atr | 1 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 64 | 4 | Atr | 1 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 75 | 5 | Atr | 1 | 12 | 264 | 242 | 270 | 221 | 231 | 292 | 275 | 206 | 221 | 277 | |
| | 81 | 6 | Atr | 1 | 13 | 299 | 228 | 295 | 260 | 295 | 269 | 232 | 203 | 291 | 286 | |
| | 88 | 7 | Atr | 1 | 14 | 227 | 297 | 251 | 269 | 248 | 257 | 273 | 291 | 248 | 200 | |
| | 99 | 8 | Atr | 1 | 16 | 226 | 205 | 299 | 256 | 279 | 276 | 201 | 253 | 274 | 288 | |
| | 111 | 9 | Atr | 1 | 18 | 257 | 260 | 240 | 225 | 214 | 221 | 269 | 248 | 245 | 300 | |
| | 118 | 10 | Atr | 1 | 19 | 226 | 213 | 267 | 229 | 282 | 273 | 276 | 217 | 218 | 250 | |
| | 123 | 11 | Atr | 1 | 20 | 285 | 245 | 300 | 281 | 276 | 278 | 243 | 272 | 277 | 242 | |
| | 129 | 12 | Atr | 1 | 21 | 243 | 255 | 228 | 284 | 223 | 206 | 238 | 209 | 230 | 206 | |

Figure 3.6: Input data file

In the next section, our model of the ASU Computer Labs Phase 2 tour-schedule as well as other larger problem instances are solved – in orders of magnitude less time than the corresponding IP.

3.5 Computational Results

3.5.1 Tour-Schedule for ASU Computer Labs

The model described in Section 4 was applied to the real world example of scheduling student technicians at ASU Computer Labs. We worked closely with a scheduler using the data for shift requirements for the spring semester of 2003. The four labs combined have 195 distinct shifts that must be covered. In order to protect student privacy, student technician preferences and availabilities are generated from typical behaviors. We generated a class schedule for 50 students, which is the current number of technicians employed. The class schedules consist of either morning classes (unavailable from 0800-1200), afternoon classes (unavailable from 1200 – 1600), or evening classes (unavailable from 1600-2000). Each student technician is assumed to attend a typical student-worker load of three different classes per week, with each class meeting twice a week. Therefore, each student is unavailable for six time periods per week and available all weekend. We ran five distinct models, where each model adds one more of the real world constraints described in Section 4. Solutions used CPLEX 8.1 on a 2.4 GHz PC with parallel processors and 1 Mb of RAM.

| Model | CPU Time (sec) | Optimal Objective |
|--|-------------------|----------------------|
| Complete Availability, Technician Preferences | 0.03 | 38625 |
| Restricted Availability, Technician Preferences | 0.04 | 38495 |
| Restricted Availability, Technician Preferences, Consecutive Shift Restrictions | 0.29 | 38452 |
| Restricted Availability, Technician Preferences, Consecutive Shift Restrictions, Varying Skill Sets | 0.70 | 38091 |
| Restricted Availability, Technician Preferences, Consecutive Shift Restrictions, Varying Skill Sets, Preference + Reward | 0.93 | 43088 |

Table 3.3: Results of the tour-schedule model for ASU computer labs

In each case, a feasible solution to the tour-scheduling problem is solved in a fraction of a second, rather than seconds or minutes. Why is this computational improvement important? In general, it means that variations on optimization problems may be run in real-time that allow the manager to see the effect of changes of interest. In the ASU Computer Lab case study, the following are examples of such uses:

- Re-rostering when someone is unavailable, or whose schedule changes (dropping and adding classes after the initial schedule, for example).
- Introduction of the student confidential information, after an initial basic schedule is built by developers.
- Determination of which types of employees are needed to improve coverage and even staff sizing (which we cover in detail in Section 5.3).

- Study the effects of rewards chosen by the manager and preferences selected by the students.

These types of analyses are essentially impossible in the manual system, yet require only seconds for a re-roster up to hours for a sensitivity analysis.

These time periods are a small fraction of the 2-3 days spent just to get one feasible schedule manually using the current practice. They are also a fraction of the time needed if solutions were obtained by a brute-form IP as we can see from Table 3.4. The benefits of our method increase as problem size grows larger. The ASU example has about 10,000 binary variables and 1,000 constraints, but is in no way a challenge to our method. In the next sub-section we look at the computational behavior of larger problem instances. Those larger problems are randomly generated problems, not from our case study, but containing the same types of features.

3.5.2 Computational Efficiency for Larger Problem Instances

Four larger problem instances were solved using realistic data. The computational effort is summarized in Table 3.4 below. The ASU computer lab problem is on the order of the third problem listed. Our formulation continues to solve the tour-scheduling problem in an insignificant amount of computer time even for very large problem instances of 420 distinct shifts (about the number of shifts in three months with a granularity of 4 hours on shift definition) and 80 employees. Our formulation is completely adaptable to scheduling additional time periods and, as we can see below, the formulation will solve these larger problems very quickly.

| Number of Shifts | Number of Employees | Network-Based LP | | | Integer Program | | |
|------------------|---------------------|------------------------|-------------|----------------|--------------------|-------------|----------------|
| | | Variables (continuous) | Constraints | CPU Time (sec) | Variables (binary) | Constraints | CPU Time (sec) |
| 35 | 10 | 420 | 205 | 0 | 350 | 90 | 0.02 |
| 70 | 20 | 1540 | 430 | 0 | 1400 | 180 | 0.07 |
| 175 | 50 | 9100 | 1225 | 0.06 | 8750 | 450 | 1.53 |
| 420 | 100 | 42700 | 3020 | 0.5 | 42000 | 1040 | 135.98 |

Table 3.4: Computational results for larger problem instances

Figures 3.7 and 3.8 illustrate the emerging gap in computational time required to get a solution between our formulation, Network-based Mathematical Program (NBMP), and a brute-force IP. Our formulation will be able to solve much larger problems before the computational time becomes excessive. This will allow businesses to schedule months, quarters, or years at a time.

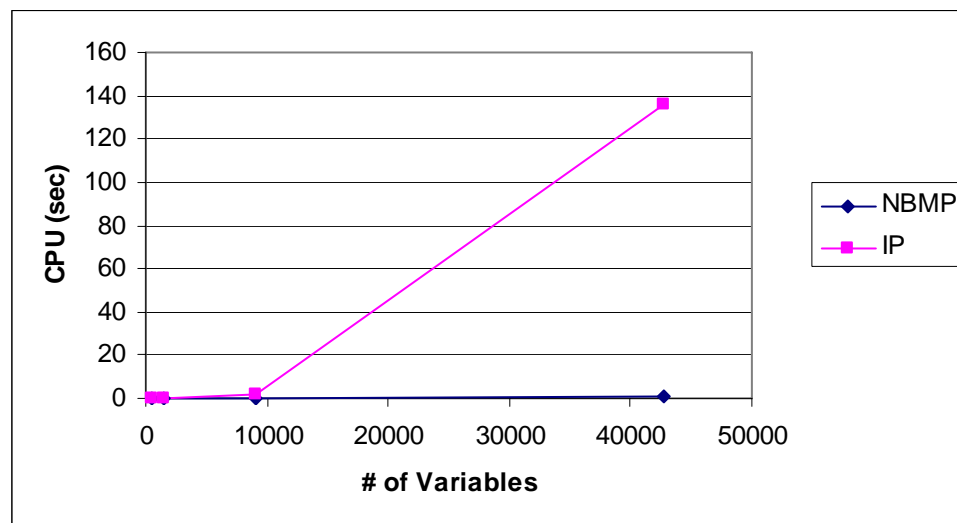


Figure 3.7: Computational time comparison versus the number of variables

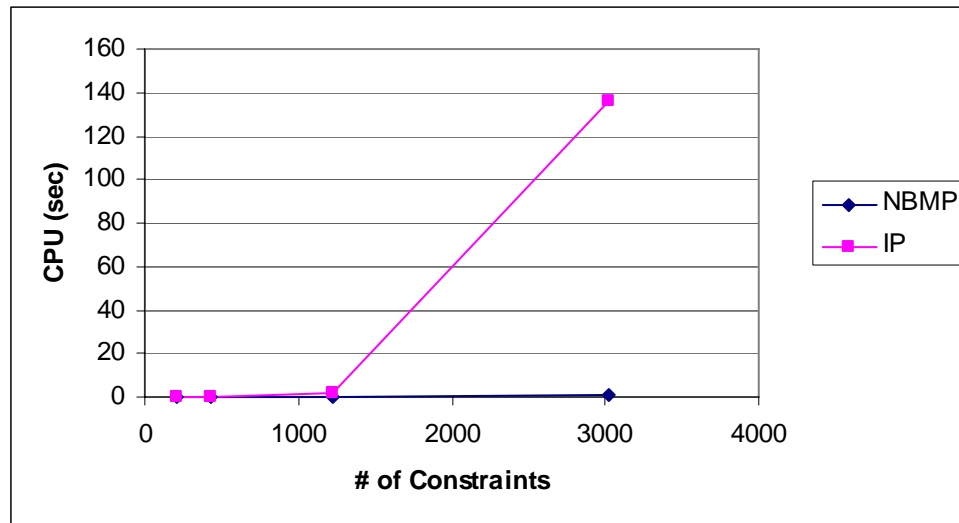


Figure 3.8: Computational time comparison versus the number of constraints

3.5.3 Staff Sizing Application

The speed at which the formulation can be solved and the ease with which real world constraints such as employee availability, consecutive shift restrictions and varying skills sets are incorporated, allows for an additional important result for industries with heterogeneous workforces. Industries that are looking to streamline their employee payrolls or are opening a new facility can use the algorithm to predict the minimum number of staff required to fill shifts based on customer demand.

Given a level of employee availability, the industry standard for rest between consecutive shifts, and the number of skills required, a set of employee data can be generated. The formulation can then be solved for this case and the minimum number of employees that covers that definition of demand can be found by running the model until a feasible solution. Repeating this for several

sets of employee data, an actual minimum number of employees needed to staff the business will become apparent.

Below is an example of how our formulation can be used to generate the minimum number of employees required. The example problem instance contains 49 shifts (7 shifts per day for a week) and 15 employees. The table shows employee requirements for a typical industry as employee availability mix reduces from full (100%) availability to half time (50%) availability. The availabilities are generated randomly for this example, so 50% availability would mean each employee is randomly unavailable for half of the hours each week. In Table 3.5, below each availability is the minimum number of employees required for that problem instance. An average, standard deviation, and a minimum employee to shift ratio is calculated.

| Availability | | 100% | 90% | 80% | 70% | 60% | 50% |
|--------------------------------------|---------|-------|-------|-------|-------|-------|-------|
| Minimum Number of Employees | Trial 1 | 10 | 11 | 12 | 13 | 15 | 15 |
| | Trial 2 | | 12 | 13 | 14 | 14 | 15 |
| | Trial 3 | | 11 | 12 | 14 | 14 | 16 |
| | Trial 4 | | 11 | 13 | 14 | 15 | 15 |
| Average Minimum | | 10 | 11.25 | 12.5 | 13.75 | 14.5 | 15.25 |
| Standard Deviation | | | 0.500 | 0.577 | 0.500 | 0.577 | 0.500 |
| Employee/Shift Ratio | | 0.204 | 0.230 | 0.255 | 0.281 | 0.296 | 0.311 |

Table 3.5: Employee requirements as a function of employee availability

As expected, as the time each employee has available decreases the minimum number of employees required to staff the demanded number of shifts increases. This formulation leads to an employee/shift ratio. Given the parameters of the proposed business, the entrepreneur can look at varying

scenarios, chose as conservative an estimate as desired and find the ratio of employees to shifts from a graph like the one shown below. If circumstances change as the business moves closer to opening or in streamlining processes, the graph can be consulted to determine a new level of staffing ratio based upon any new level of employee availability. Similar graphs could be constructed varying consecutive shift restrictions or skill sets.

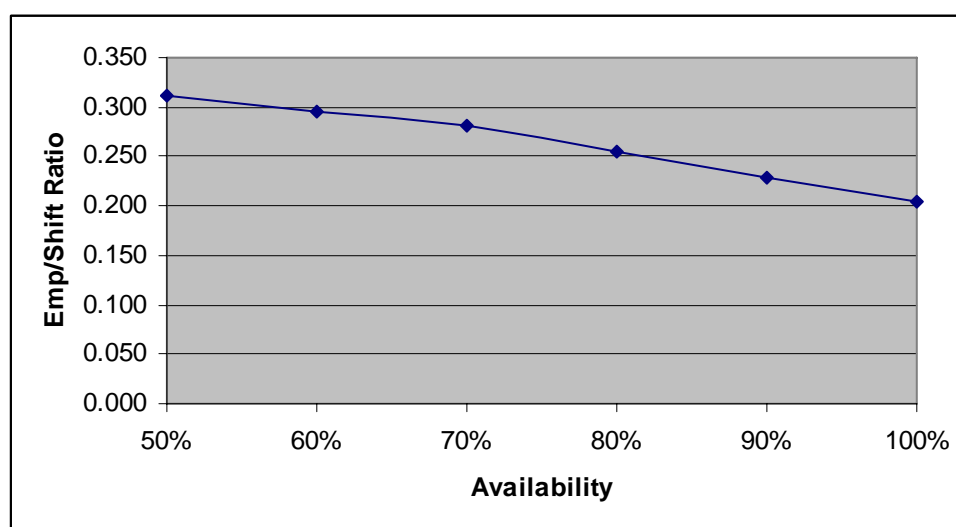


Figure 3.9: Employee to shift ratio based upon employee availability

3.6 Conclusions and Recommendations

Our formulation of the tour-scheduling problem allows a scheduler to incorporate many of the real world constraints inherent in an implementable tour-schedule. The scheduler can consider heterogeneous employee availabilities, varying skill sets, consecutive shift restrictions, and seniority or job performance incentives. Additionally, the formulation solves very large problem instances in a computationally insignificant amount of time compared to the corresponding IP

solution. This allows the scheduler to reschedule often or consider new information that would require additional tweaking of the tour-schedule.

The formulation was applied successfully to a real world example involving a very heterogeneous workforce and continuous operations. The Technology Support Analyst Principal (TSAP) for the ASU computer labs indicated that it took him 2 to 3 days per semester to schedule the computer labs. Due to the high constraint on the availabilities of the technicians it is a very laborious process that the TSAP does manually with the aid of a spreadsheet for recording the tour schedule. By hand, it is very difficult to ensure that students are given a proper amount of rest between shifts. Shifts that occur consecutively or within a minimum number of hours are deemed to be conflicted and not scheduled to the same technician. The proposed model can accommodate any minimum window of rest required by the scheduling organization and can, therefore, schedule businesses with continuous operations.

Finally, we took advantage of the computational efficiency of the model and solved many problem instances to generate a minimum employee to shift ratio. Such an analysis would be very valuable to any business seeking to cut payroll or to an entrepreneur who is looking to open a new business and needs to determine appropriate staffing levels dynamically as conditions change, or in the planning stages of a business that does not exist yet.

In all cases we considered, the inclusion of heterogeneous workers and mandated rest breaks in a continuous schedule add realism to the tour scheduling problem. Appendices A and B contain the code and CPLEX output.

Chapter 4

A MATHEMATICAL PROGRAMMING APPROACH TO OPTIMAL NURSE ROSTERING OVER CONTINUOUS AND MULTI-WEEKLY TIME PERIODS WITH EMPLOYEE WEEKEND RESTRICTIONS AND SHIFT PREFERENCES

Abstract

This paper demonstrates an efficient mathematical programming approach to optimally solving the nursing tour-scheduling problem for continuous (24-hour) hospital operations and across multi-weekly time periods. The tour-scheduling problem is extended from the traditional weekly schedule to monthly, quarterly, and yearly schedules. The network model is amended to contain a nodal structure that will limit the number of weekends each employee is required to work per month. Employee shift preferences and managerial employee preferences are used as the objective function and are maximized to optimality with this approach. The framework is very flexible, it allows for a heterogeneous workforce, varying wage rates and skill levels, employee and management preferences, and continuous (24-hour) operations. The approach is validated using a practical nurse scheduling problem instance for Banner Health Hospitals, a leading health care provider in Phoenix, Arizona and the surrounding areas.

4.1 Introduction

Personnel scheduling is one of the most difficult, important, and often studied problems in operations research. The optimal choice of the number of employees required to meet customer demand, shift start and stop times, daily lunches and breaks, and the assignment of the adequately skilled employee to the best shift is a large combinatorial optimization problem. Managers who attempt to manually solve the personnel scheduling problem expend many valuable work hours to find even a feasible solution which has little probability of being optimal based on any objective function. Sub-optimal scheduling increases an industry's tangible costs, not only through the consumption of a manager's time, but also through the misallocation of shifts to meet customer demand and through employees staffing shifts for which they are not qualified. Furthermore, a sub-optimal schedule will increase intangible costs such as lower employee morale from dissatisfaction with a poor work schedule. Eliminating these costs is an easy way to increase the profitability of any business.

Personnel scheduling is a necessary chore associated with any organization employing people to get work accomplished. Some organizations that employ a professional workforce during traditionally scheduled business hours find the personnel scheduling problem of trivial concern. However, industries that rely on a large workforce with differing skill sets, highly constrained availabilities (work hours), and require staffing for the entire day and evening hours or even continuous (24-hour) operations have a much more

challenging problem. One such industry is the health care industry, specifically the personnel tour-scheduling problem associated with nurses.

A timely and optimal solution to the personnel scheduling problem will save each manager valuable time, meet customer demand, and increase worker morale by matching each employee to the best shift possible. This article presents an extremely fast, computationally efficient, and optimal network-flow based mathematical programming solution for the extremely challenging problem of rostering a nursing workforce. We will demonstrate that the formulation can find integer solutions without branching, bounding, or cutting schemes.

4.1.1 Problem Overview

The large, complex problem of personnel scheduling is generally decomposed into more tractable sub-problems. The three primary sub-problems are demand modeling, shift selection, and employee tour-scheduling or rostering. Demand modeling determines the number of employees required during a given time interval to satisfy customer needs. Forecasting and queuing theory are the primary tools applied to demand modeling.

The second primary personnel scheduling sub-problem is shift selection. Shift selection assigns consecutive hour shifts to satisfy the customer demand while meeting organizational and regulatory requirements for shift lengths, breaks, and mealtime allowances. The objective of shift selection is generally to minimize the number of hours scheduled that exceed customer demand.

Finally, the third primary personnel scheduling sub-problem is tour-scheduling also known as employee rostering. Tour-scheduling assigns

individuals to specific shifts detailed during shift selection. A heterogeneous workforce is a collection of personnel who have significantly variant availabilities, skill sets, and wage rates. Tour-scheduling a heterogeneous workforce must consider all of these differences and optimally match the personnel with the best shift possible and for which they are eligible. Additionally, considerations for individuals working a minimum and maximum number of shifts per week and insuring adequate rest between shifts for industries with continuous (24-hour) operations are critical to an optimal tour-schedule.

Additionally, a tour-schedule that considers employee shift preferences and management employee weighting is very advantageous to an organization. If employees are matched to shifts that fit well with their home life, availabilities, work habits, or other personal considerations, they will be motivated to work harder and more efficiently. The framework to collect and represent these preferences must be repeatable and comprehensible to the workforce.

The formulation and methodology presented in this article provide several contributions to the nurse scheduling literature. The following list summarizes the advancements proposed by this research.

- Provide a formulation that can account for variations in nursing skill sets and assign them to appropriate shifts for which they have the proper training
- Insure adequate rest between shifts and limit the number of weekends assigned per month
- Efficiently roster for monthly, quarterly, or yearly schedules

- Describe a structured framework for obtaining employee shift preferences and managerial preferences for each employee

This article is organized in the following manner. Section 2 contains a survey of the pertinent literature and important previous work accomplished in nursing tour-scheduling. Section 3 details the nurse tour-scheduling problem at Banner Health. Our methodology is general, but we feel that for realism and clarity we will introduce our case study early. Section 4 provides the proposed solution method and model. Section 5 shows the results of the methodology as applied to the case study. Section 6 finishes with conclusions and recommendations.

4.2 Literature Survey

One of the most studied applications in personnel scheduling is nurse scheduling. Nurse tour-scheduling or rostering is a subset of the literature for this application. The literature on nurse tour-scheduling is characterized by a known demand for nurses and a given set of nurse shifts. The heterogeneous workforce is characterized by varying nurse seniorities or skill sets and continuous or 24-hour operations. Ernst, et. al. (2004) give a breakdown of the various algorithmic approaches to nurse tour-scheduling. These mathematical approaches include simulation, heuristics, global search techniques, constraint programming, and mathematical programming. The scope of this literature survey will be limited to the mathematical programming approaches, the

differences between these approaches, and the proposed model in this research which I term the network-based mathematical program (NBMP).

Jumard, et. al. (1998) propose a 0 - 1 mixed integer linear program (MIP) master problem to solve the nurse tour-scheduling problem. They incorporate a heterogeneous workforce, based upon nursing specialties or skills, overlapping shifts, and nurse preferences. The master problem contains a set of feasible shifts for the planning horizon that meet all the constraints. The goal of the master problem is to match a nurse with each shift pattern. Other constraints such as the minimum number of nurse with skill level ℓ scheduled at any time and the ratio of less experienced to more experienced nurses are contained in the master problem. A branch and bound approach is taken to solve the master problem and maximize the preferences of the nurses. After the initial linear programming relaxation, an auxiliary problem is then solved using a resource constrained shortest path. The auxiliary problem determines where to branch and which column to generate into the basis.

Similarly, Millar and Kiragu (1998) develop patterns of shifts and days-off, which they term stints, where “all possible work-stretches can be pre-specified”. They attempt to find a set of stints that cover the demand for nurses using a mixed-integer branch and bound strategy. Warner (1976) also proposes an approach where all feasible patterns of shifts and days off were enumerated and then allocated to a nurse.

4.3 Nurse Tour-Scheduling Problem

Banner Health is a large health care provider in Phoenix, Arizona and the surrounding areas. Banner runs many different hospitals that specialize in many different aspects of health care. One commonality from hospital to hospital is the need for tour-scheduling its cadre of nurses. The cadre is made of many different specializations, different daily staffing levels, and hours of operation. We have chosen a representative case study on which to apply our methodology. This case study will be used to illustrate our proposed solution technique. We do this without the loss of generality to other problem instances within the health care field.

4.3.1 A Case Study: Banner Health Baywood Hospital

We relied on the expertise of James Broyles, an industrial engineer at one of Banner Health hospitals, to help us frame the problem as realistically as possible. Actual nurse preferences were not obtained; however, the production questionnaires or surveys that would have been needed to acquire such preferences are not important to the illustration of the methodology. Representative preferences were generated using the framework proposed in Section 4.4.

The Intensive Care Unit at Banner Health Mesa is staffed with qualified nurses 24 hours a day due to the nature of care required in a critical care situation. The number of nurses required each day varies with day of the week and month of the year. We will pick a representative number of 3 distinct shifts per day. Each distinct shift will be staffed by multiple nurses. The shifts will have

staggering lunch hours and break times to insure adequate staffing. The determination of these shifts (shift selection sub-problem) has been done by Banner Health Baywood and is not part of the scope of this article. We will staff a cadre of both 20 and 40 nurses which will illustrate both a slow and heavy workload and develop a weekly schedule. Each nurse will not work more than 48 hours per week or fewer than 24. This will represent the part-time and full-time mixture of nurses at Banner Health Baywood.

Finally, we will expand our model to efficiently schedule multi-weekly time periods. We will demonstrate the model's ability to schedule bi-weekly, monthly, quarterly, and even yearly tour schedules. An important consideration when scheduling nurses over multi-weekly schedules is the number of weekends each nurse will work per month. We can easily amend our model with a nodal structure that can insure the restriction on weekend shifts is met.

4.4 Proposed Tour-Schedule Solution Method and Model

The tour-scheduling problem is inherently a binary set-covering problem. Therefore, as problem size increases, the efficiency of the associated integer program decreases rapidly. However, we have developed a formulation of the problem as a minimum cost network-flow, using an arc capacity method, so that the resulting network structure generally provides integer binary answers. The pseudo-network structure can easily be written as an LP and solved using fast solution algorithms such as the CPLEX interior point method. Many solutions of this formulation do not require any branching, bounding, or cutting schemes to

find integer solutions – the integer solutions are a consequence of the formulation method. Therefore, very large problem instances can be solved in a computationally insignificant amount of time compared to the corresponding IP solution. The data will empirically show that even when solutions to the problem using our formulation are not entirely integer, they are high quality solutions that provide all but a small percentage of fractional values. The convex hull of the NMBL formulation closely coincides with the optimal solution from a pure integer mathematical program. Only cutting techniques automatically implemented within CPLEX are needed to find the integer solution extremely efficiently, with little to no extra computational time. No branching is needed due to the nearly optimal characteristic of the solution to the network-based mathematical program. Figure 4.1 shows the formulation of our generalized weekly tour-scheduling problem as a minimum cost network-flow.

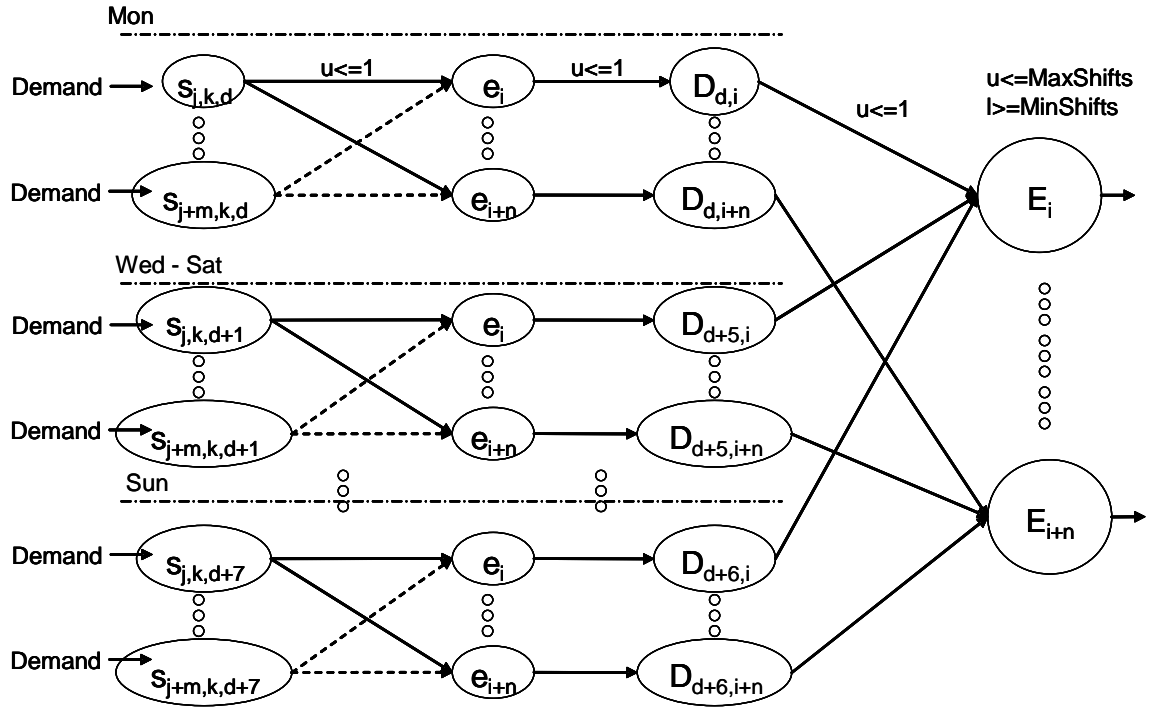


Figure 4.1: Network representation of the Phase 2 tour-scheduling problem

Here, $s_{j,k,d}$ denotes shift number j requiring skill set k and on day d , e_i is employee number i , $D_{d,i}$ is the total number of shifts, D , on day d for employee i , and E_i is the total number of shifts per week, E , for employee i .

The number of employees needed for each shift node flows from the Demand arc. Each shift node has an edge to each qualified and available employee node. The shift-to-employee arc is capacitated at 1, meaning only one of the required staffing for a shift can be assigned to a single employee. Each employee then flows their daily work assignment to the employees' daily-total-node, D . This new arc is capacitated at 1, meaning each employee can work only one shift per day. Finally, the employees' daily-total-nodes channel the flow

to the employees' weekly-total-node which contains the capacities to enforce minimum and maximum shifts per week.

The weekly tour-scheduling model above is then stacked upon itself to simultaneously schedule longer time horizons, such as months, quarter, or years depending upon the requirements of the industry. Figure 4.2 illustrates this extended problem.

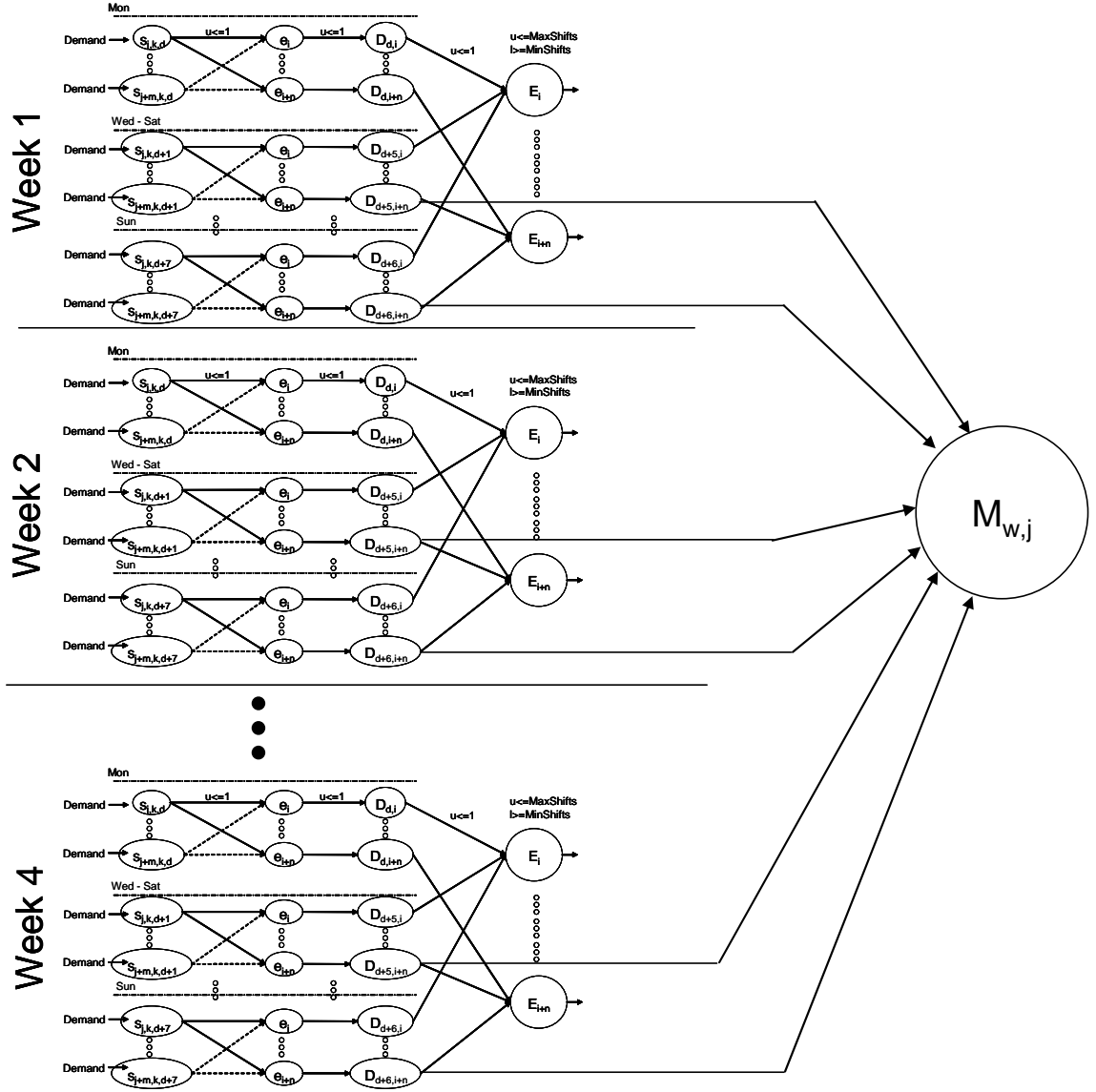


Figure 4.2: Multi-weekly network representation of tour-scheduling problem

Here $M_{w,j}$ is the total number of weekend days, w , worked in a month for each employee j . The node is capacitated at the number of weekend days each nurse is allowed to work each month. Therefore, we must add a set of constraints, equation 4.6, to the mathematical formulation given in Chapter 3's article, "A

Network-based Mathematical Programming Approach to Optimal Rostering of Continuous Heterogeneous Workforces”.

The balance equations for the multi-weekly network flow formulation are:

$$\sum_i s_{j,k,d} e_i = demand \quad \text{for all } j,k,d \quad (4.1)$$

$$\sum_{j,k} s_{j,k,d} e_i = D_{d,i} \quad \text{for all } i,d \quad (4.2)$$

$$\sum_d D_{d,i} \leq \max number_of_shifts \quad \text{for all } j \quad (4.3)$$

$$\sum_d D_{d,i} \geq \min number_of_shifts \quad \text{for all } j \quad (4.4)$$

$$D_{d,i} \leq 1 \quad \text{for all } i,d \quad (4.5)$$

$$\sum_w W_{m,j} \leq \max weekend_shifts \quad \text{for all } j \quad (4.6)$$

Constraint set 1 requires that the demand for a number of employees for each type of shift is met. Set 2 totals the shifts per employee for each day and requires that the total be less than or equal to 1 in constraint set 5 (no employee works more than one shift per day, but remember that the shift length is a variable). Constraint sets 3 and 4 require that the minimum and maximum numbers of shifts per week are met for each employee. Finally, set 6 restricts the number of weekend days worked by each employee. The following three sub-sections describe how our model incorporates all the constraints needed to create a tour-schedule for a real world health care organizations, including a heterogeneous work force and consecutive shift restrictions.

4.4.1 Consecutive Shift Restrictions

Adequate rest between consecutive shifts is a requirement for any realistic tour-schedule of continuous operations. Industries that are only open for an 8-

hour day have tour-schedules that implicitly contain adequate rest. However, when scheduling an industry, such as ICU nurses, with continuous or 24-hour operations, restrictions on consecutive shifts is a requirement for any realistic tour-scheduling formulation. Love and Hoey (1990) and Loucks and Jacobs (1991) note the need for consecutive shift restrictions and state the lack of such restrictions as limitations to their tour-scheduling models. Our formulation allows for specialized side constraints to accommodate continuous operations.

As input in any specific application, we use the appropriate industry standard for how much time is required between the start of shift j and the start of the next shift. This leads to a matrix of shifts, J , that conflict with each other in the sense that an employee assigned to work shift j cannot be scheduled to work any shift conflicted with shift j . The following constraint is added to constraints 1-5 above to accommodate consecutive shift restrictions.

$$\sum_{j \in J} s_{j,k,d} e_i \leq 1 \quad \text{for all conflicted shifts} \quad (4.7)$$

Our empirical experience indicates that in many cases these side constraints do not destroy the integrality of the solution to the minimum cost network flow.

4.4.2 Employee Availabilities and Varying Skill Sets

Many industries have employees with heterogeneous availabilities. For example, service industries often employ younger, part-time employees who have restrictions on their time due to schooling or other activities. Our model can be modified to accommodate such heterogeneous availabilities. We simply remove the network arc from a shift to an employee who is not available during

the hours encompassing that shift. No flow units can travel from that shift to the employee and consequently the employee is not assigned to that shift.

Similarly, it is often the case that many service industries have different skill requirements needed during a particular shift. This is especially true of the nursing industry where nurse often specialize in different level of care or surgical-assistance expertise. If a nurse does not possess the required skill for a particular shift, the arc from the shift to the employee is removed from the network. If the nurse has multiple skill sets, the arc from the employee to all shifts within their set is contained within the network.

This is an important aspect to the efficiency of the tour-schedule. Many examples within the tour-scheduling literature schedule only one department within the industry at a time. Therefore, the employees assigned to that department can be scheduled only within that department and there is no mechanism to schedule that employee across departments. This could lead a department to over-staffing costing a business extra payroll. We have no such restriction.

Removal of the two types of arcs described above are easy to program. They have no effect on the integrality of solutions from the LP-network solution. This is because the resulting network is still of minimum cost flow form.

4.4.3 The Objective Function - Employee and Manager Preferences

The objective function of the constraint set described above can now be defined. The objective function is made up of the preferences for each shift from the employees and by the ranking of the employees from the manager. It should

be noted that there are two distinct situations when the formulation presented may cause the solver to find an alternate optimal solution that contains fractional employee assignments (there is always an integer solution). The first situation can be subverted through the intelligent use of preferences. This is discussed next and the second is discussed in the section 4.5 Computation Results.

Fractional solutions may occur when the preferences of a set of employees for a particular shift are equal and none is violating a consecutive shift restriction, maximum number of shifts per week constraint, or have more than the allotted weekend shifts. The solution is obviously an alternate optima, since the entire shift can be given to any of the fractional employees and the objective function will not change, nor is any constraint violated. This phenomenon can be combated by using relative rankings for both the shifts and the employees.

Relative rankings will break ties between the shift preferences of employees. Each employee, j , ranks all the shifts and each shift is given a score, $emppref_{i,j}$, on a relative scale, 1 through the number of shifts, ensuring no employee has the exact same preference for each shift, i . Next, the manager ranks all employees and then each employee is given the appropriate relative score, $manpref_i$, ranging from 1 through number of employees. This number is then given an order of magnitude increase, O , over the scale for employee shift preferences and the two preferences are added together. The preferences of each shift/employee combination are the aggregate of the employee's and manager's preferences and are distinct from one another.

$$pref(s_{j,k,d}e_i) = emppref_{i,j} + (O \bullet manprep_i) \quad (4.8)$$

The objective function for the constraint set described in equations 4.1 – 4.6 then becomes

$$\text{MAX} \quad \sum_{j,k,d,i} s_{j,k,d} e_i \bullet \text{pref}(s_{j,k,d} e_i) \quad (4.9)$$

Figure 4.3 graphically illustrates the employee/shift preferences for the first four shifts and four employees after applying the above approach. The most important employee, as ranked by the manager, is on the top of the graph.

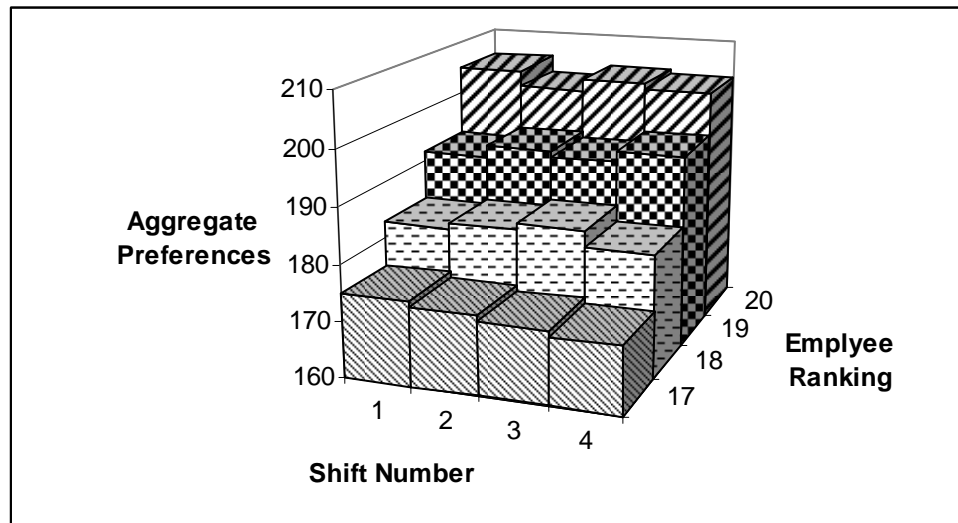


Figure 4.3: Illustration of aggregate preferences for employ/shift combination

Data organization to support these calculations is straightforward. A spreadsheet is used as the input data file. A Visual Basic macro then writes the linear program in CPLEX .lp input format. This macro runs almost instantaneously. The output file is then given to CPLEX for optimization. The ease of changing data in the input data file, the speed with which the output file is written, and the speed with which the problem is optimized (more on this in the next section) allows the user to study a wide variety of problem instances and

conduct sensitivity analysis. Figures 4.4 and 4.5 show examples of the input data files.

| Employee Shifts | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|-----------------|-------------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | Preferences | | | | | | | | | | | | | | | | | | | |
| 1 | 2175 | 2022 | 1875 | 1933 | 1824 | 1656 | | 1489 | 1430 | 1265 | 1221 | 953 | 935 | 923 | 815 | | | | 205 | 280.5 |
| 2 | 2206 | | | 1917 | | | 1632 | 1472 | | 1314 | | 989.5 | 982.5 | 783.5 | 615 | | 575.5 | | 439.5 | 233 |
| 3 | | 2094 | | | | 1590 | | | | 1259 | | 984 | | 742 | 837.5 | | | 356.5 | 354.5 | |
| 4 | | 1926 | 2006 | | 1741 | 1560 | 1457 | 1445 | | 1330 | 1237 | 1006 | 949 | 900.5 | | | 624.5 | 492 | 320.5 | 180 |
| 5 | 2210 | | | | | | 1549 | 1425 | 1358 | 1177 | | 955.5 | | 916.5 | 636.5 | 707 | 464.5 | 432.5 | 404.5 | 316 |
| 6 | | 2044 | 1909 | | | | 1612 | | | | | 994.5 | 1025 | | 751.5 | | 587 | 313 | 375 | 125 |
| 7 | 2246 | 1925 | 1979 | 1840 | 1746 | 1629 | 1547 | 1304 | 1412 | 1290 | | 922 | 939.5 | | | | 574 | 528.5 | 351.5 | 297 |
| 8 | 2230 | | 1932 | | 1653 | | 1638 | 1372 | | 1185 | 1238 | | 850 | | 759.5 | 703.5 | 642.5 | 350 | 429.5 | |
| 9 | 2059 | 1907 | 1901 | | 1847 | | 1508 | 1420 | 1398 | | 1106 | 1121 | 1048 | 935 | | 553 | 610.5 | 498 | 398.5 | 166.5 |
| 10 | | | | | 1766 | | 1536 | | | 1260 | 1264 | | 1022 | 993.5 | 944 | 842 | 649.5 | 452 | 260 | 330.5 |
| • | | | | | | | | | | | | | | | | | | | | |
| • | | | | | | | | | | | | | | | | | | | | |
| • | | | | | | | | | | | | | | | | | | | | |
| 200 | 2191 | 2105 | 1816 | | | | 1588 | 1527 | 1341 | | 1034 | 1109 | | 839.5 | 660.5 | 668.5 | 612.5 | 533 | 408 | |
| 201 | 2233 | | | 1809 | 1672 | 1703 | 1463 | 1331 | 1217 | | 1214 | 1150 | 811 | | 802.5 | 622.5 | 603.5 | | | 335.5 |
| 202 | 2237 | 2091 | | 1791 | 1750 | 1695 | | | 1273 | | 1083 | 1125 | 1027 | 826.5 | | 525 | | 376.5 | | 311.5 |
| 203 | | | | | | | 1465 | 1344 | 1247 | 1307 | | | 966.5 | | 735 | | | | 305 | 300 |
| 204 | 2239 | 2125 | 1986 | 1882 | 1675 | | 1554 | 1477 | | 1295 | 1250 | 933.5 | 852.5 | 829 | 713.5 | 641 | 583.5 | 491 | 399 | 172 |
| 205 | | 2056 | 1835 | | 1619 | 1579 | | | 1448 | 1230 | 1088 | 1136 | 1001 | 878 | | 568 | 620 | | | |
| • | | | | | | | | | | | | | | | | | | | | |
| • | | | | | | | | | | | | | | | | | | | | |
| • | | | | | | | | | | | | | | | | | | | | |

Figure 4.4: Employee/Manager shift preferences input data file

| Shift ID | Conflicted Shifts | | | | | | | | | | | |
|----------|-------------------|----|----|----|----|----|----|----|----|----|----|----|
| 1 | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | |
| 6 | | | | | | | | | | | | |
| 7 | 19 | 20 | 21 | 22 | 23 | 24 | | | | | | |
| 8 | 19 | 20 | 21 | 22 | 23 | 24 | | | | | | |
| 9 | 19 | 20 | 21 | 22 | 23 | 24 | | | | | | |
| 10 | 19 | 20 | 21 | 22 | 23 | 24 | | | | | | |
| 11 | 19 | 20 | 21 | 22 | 23 | 24 | | | | | | |
| 12 | 19 | 20 | 21 | 22 | 23 | 24 | | | | | | |
| 13 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 14 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 15 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 16 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 17 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |

Figure 4.5: List of conflicted shifts input file

In the next section, our representative model of the Banner Health nurse tour-schedule as well as other larger problem instances is solved.

4.5 Computational Results

Before presenting the computational results, the second situation where fractional decision variables may occur must be discussed. The second phenomenon where fractional solutions are returned by the solver is due to a small mismatch in the convex hull of the optimal solution to the NBMP and that of the optimal integer solution. The matrices for the larger of the problem instances are very expansive. The models being solved are on the order of tens-of-

thousands of constraints and ten-of-thousands of variables. However, because the NBMP solution is so close to the integer optimal, minimal additional computational time is required to find the integer solution. A comparison of each solution and the differences in optimal values is presented below. Each problem was solved with and without the requirement of an integer constraint on the decision variables.

| Number of Weeks | Number of Employees | Number of Shifts | Optimal Obj. NBMP | Number of Variables | Number of non-IP variables | Optimal Obj. NBMP/ w IP | Difference |
|-----------------|---------------------|------------------|-------------------|---------------------|----------------------------|-------------------------|------------|
| 1 | 20 | 63 | 7.96E+04 | 1400 | 0 | 7.96E+04 | 0.000% |
| 2 | 20 | 126 | 1.83E+05 | 2800 | 71 | 1.83E+05 | 0.025% |
| 4 | 20 | 252 | 5.03E+06 | 5600 | 327 | 5.03E+06 | 0.107% |
| 12 | 20 | 756 | 4.71E+07 | 16800 | 960 | 4.71E+07 | 0.088% |
| 52 | 20 | 3276 | 8.94E+08 | 72800 | 4723 | 8.85E+08 | 1.019% |

| Number of Weeks | Number of Employees | Number of Shifts | Optimal Obj. NBMP | Number of Variables | Number of non-IP variables | Optimal Obj. NBMP/ w IP | Difference |
|-----------------|---------------------|------------------|-------------------|---------------------|----------------------------|-------------------------|------------|
| 1 | 40 | 126 | 5.07E+07 | 5320 | 0 | 5.07E+07 | 0.000% |
| 2 | 40 | 252 | 1.17E+08 | 10640 | 238 | 1.17E+08 | 0.029% |
| 4 | 40 | 504 | 2.51E+08 | 21280 | 574 | 2.51E+08 | 0.040% |
| 12 | 40 | 1512 | 7.85E+08 | 63840 | 1802 | 7.85E+08 | 0.039% |
| 52 | 40 | 6552 | 3.44E+09 | 276640 | 7474 | 3.39E+09 | 1.490% |

Table 4.1: Summary of results for formulation with and without IP variables

It can be seen that when developing a weekly schedule comprised of 20 employees and 63 shifts and 40 employees and 126 shifts, the NBMP formulation does not affect the ability of the model to find integer solutions without integer constraints in the constraint set. As the model grows in size a relatively small number of variables return as fractions. When scheduling 20 nurses for two weeks, only 71 of 2,800 decision variables are fractional and at four weeks only 327 of 5,600. Additionally, the last column shows the difference in the objective function value, in both the smaller cases as less than 0.107%.

Next it is shown that the model approaches the integer optimal.

Therefore, even when all variables are restricted to be integer, the solution times are in hundredths of seconds using CPLEX 8.1 on a 2.4 GHz PC with parallel processors and 1 Mb of RAM. In fact, CPLEX does not have to use any branching techniques. Figure 4.6 shows the CPLEX header output for the 4 week model. No branching is required; the only technique needed is a Gomory fractional cut.

```

Problem 'w4e20ip.lp' read.
Read time = 0.07 sec.
Tried aggregator 1 time.
MIP Presolve eliminated 21926 rows and 7009 columns.
MIP Presolve modified 319 coefficients.
Aggregator did 12 substitutions.
Reduced MIP has 1857 rows, 2035 columns, and 7354 nonzeros.
Presolve time = 0.06 sec.
Clique table members: 1817
MIP emphasis: balance optimality and feasibility
Root relaxation solution time = 0.03 sec.

      Nodes
      Node Left  Objective  IInf  Best Integer  Best Node  ItCnt  Gap      Variable B Parent  Depth
0  0  299531.8750  58          299531.8750  520
*      299458.0000  0  299458.0000  Cuts: 33  527  0.00%

Clique cuts applied: 5
Gomory fractional cuts applied: 2

Integer optimal solution: Objective = 2.9945800000e+05
Solution time = 0.13 sec. Iterations = 527 Nodes = 0

```

Figure 4.6: CPLEX header output

Table 4.2 shows that the computational time for the network-based mathematical program approach with integer constraints on the decision variables are close to or better than the pure network flow model.

| Number of Weeks | Number of Employees | Number of Shifts | Number of Constraints | Number of Variables | Solution (sec) NBMP | Solution (sec) NBMP/ w IP |
|-----------------|---------------------|------------------|-----------------------|---------------------|---------------------|---------------------------|
| 1 | 20 | 63 | 1805 | 1400 | 0.02 | 0.03 |
| 2 | 20 | 126 | 3568 | 2800 | 0.05 | 0.17 |
| 4 | 20 | 252 | 7172 | 5600 | 0.13 | 2.14 |
| 12 | 20 | 756 | 20835 | 16800 | 1.56 | 10.87 |
| 52 | 20 | 3276 | 92600 | 72800 | 7.42 | 204.15 |

| Number of Weeks | Number of Employees | Number of Shifts | Number of Constraints | Number of Variables | Solution (sec) NBMP | Solution (sec) NBMP/ w IP |
|-----------------|---------------------|------------------|-----------------------|---------------------|---------------------|---------------------------|
| 1 | 40 | 126 | 3522 | 5320 | 0.1 | 0.09 |
| 2 | 40 | 252 | 6988 | 10640 | 0.35 | 0.9 |
| 4 | 40 | 504 | 13676 | 21280 | 0.9 | 3.38 |
| 12 | 40 | 1512 | 40972 | 63840 | 8.34 | 31.26 |
| 52 | 40 | 6552 | 176781 | 276640 | 20.65 | 430.51 |

Table 4.2: Computational time results

Finally, we compare our results with those from a traditional MIP originally computed in the article in Chapter 3 entitled “A Network-based Mathematical Programming Approach to Optimal Rostering of Continuous Heterogeneous Workforces”. Figure 4.7 compares the computational time of a traditional tour-scheduling mixed-integer programming formulation with the network-based mathematical programming approach both with and without integer constraints. We can see that the NBMP out-performs the more traditional approach as the problem size grows larger.

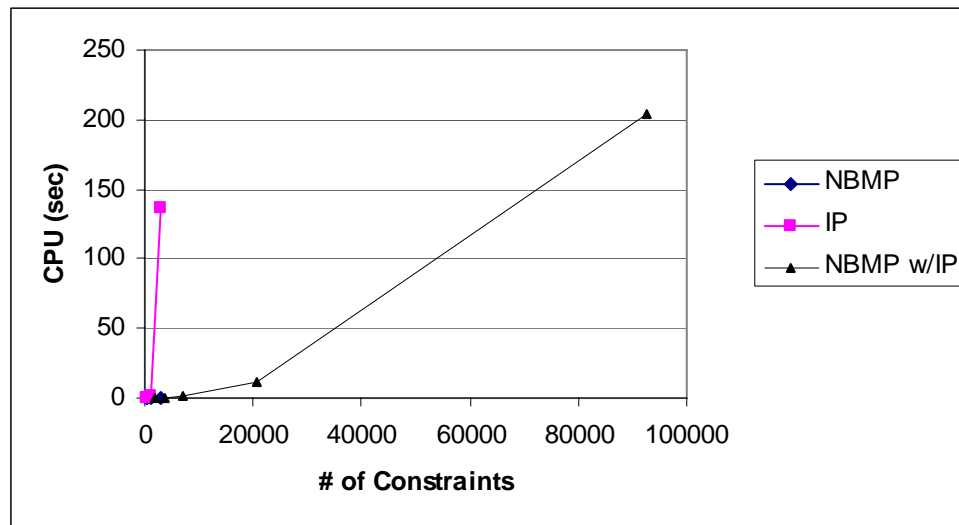


Figure 4.7: Comparison of computational times and number of constraints

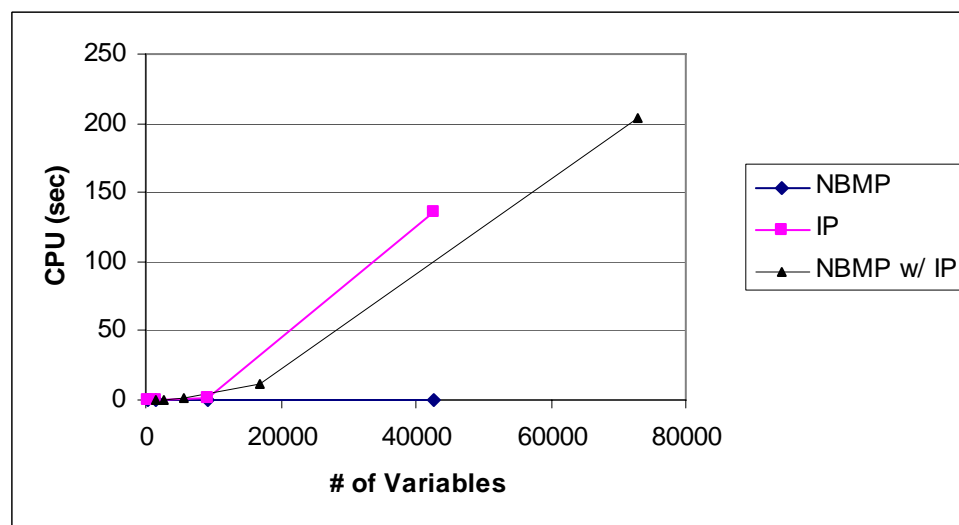


Figure 4.8: Comparison of computational times and number of variables

Finally, illustrated in Figure 4.8, similar efficiency of the NBMP as the variables of the model are increased is shown.

4.6 Conclusions

In this article we demonstrated a network-based mathematical programming approach to solving the tour-scheduling problem. We validated our approach by solving a realistic nurse rostering problem instance at Banner Health Baywood Hospital in Phoenix, AZ. The case study is scheduled at two different staffing levels and across multiple weeks, including up to one year. Next, we present a repeatable and easily understood framework for collecting employee shift preferences and managerial employee rankings. These preferences are then maximized to optimality. Finally, we compared the efficiency of the network-based mathematical approach to a traditional tour-scheduling mixed-integer program. This comparison illustrates the minimal computational time NBMP requires as the problem grows larger. The network-based mathematical programming approach proposed in this article is a valid, efficient, and optimal approach to solving the personnel tour-scheduling problem.

Appendices C contains the Visual Basic code used to write the model into a CPLEX input file. Appendix D contains all the CPLEX output for the nursing models solved for this article.

Chapter 5

A NETWORK-BASED MATHEMATICAL PROGRAMMING APPROACH TO USING EMPLOYEE PREFERENCES IN RE-ROSTERING OPTIMAL TOUR- SCHEDULES

Abstract

This article presents a methodology that will efficiently re-roster an optimal schedule when employee absenteeism causes a shift to be re-staffed. The goal and the success of the methodology presented are to minimize the disturbance to the original schedule while covering all unfilled shifts. Employee preferences, needed to find the original optimal schedule, are deftly modified to ensure the minimization of the disturbances. The manager is required to gain no new information and can quickly publish a new schedule, thus saving valuable time and notifying the changes to the workforce as soon as possible. The re-rostering technique is then validated by re-rostering two optimal schedules produced for two real-life case studies: Arizona State University Computer Labs and Banner Health Baywood Hospital. These case studies are large personnel tour-scheduling problem instances and each are scheduled under three distinct scenarios. The scenarios show the performance of the methodology when re-rostering is required in benign, general, and difficult conditions. The methodology requires hundredths of a second of computational time and minimizes the deviation to the original schedule.

5.1 Introduction

Personnel scheduling is one of the most difficult, important, and often studied problems in operations research. The optimal choice of the number of employees required to meet customer demand, shift start and stop times, daily lunches and breaks, and the assignment of the adequately skilled employee to the best shift is a large combinatorial optimization problem. Managers who attempt to manually solve the personnel scheduling problem expend many valuable work hours to find even a feasible solution which has little probability of being optimal based on any objective function. Sub-optimal scheduling increases an industry's tangible costs, not only through the consumption of a manager's time, but also through the misallocation of shifts to meet customer demand and through employees staffing shifts for which they are not qualified. Furthermore, a sub-optimal schedule will increase intangible costs such as lower employee morale from dissatisfaction with a poor work schedule. The problem is further complicated by employee sicknesses, vacations, hiring, or firing. In this inevitable instance the sub-optimal problem must again be adjusted, costing more hours, to find another feasible schedule that will also likely be sub-optimal. This article presents a methodology to solve the problem of re-rostering when this employee absenteeism occurs.

5.1.1 Problem Overview

A primary personnel scheduling sub-problem is tour-scheduling also known as employee rostering. Tour-scheduling assigns individuals to specific shifts detailed during shift selection. A heterogeneous workforce is a collection of

personnel who have significantly variant availabilities, skill sets, and wage rates. Tour-scheduling a heterogeneous workforce must consider all of these differences and optimally match the personnel with the best shift possible and for which they are eligible. Moreover, considerations for individuals working a minimum and maximum number of shifts per week and insuring adequate rest between shifts for industries with continuous (24-hour) operations are critical to an optimal tour-schedule.

A tour-schedule that considers employee shift preferences and management employee weighting is very advantageous to an organization. If employees are matched to shifts that fit well with their home life, availabilities, work habits, or other personal considerations, they will be motivated to work harder and more efficiently. The framework to collect and represent these preferences must be repeatable and comprehensible to the workforce. The original tour-schedule often can not be implemented because of employee sickness, vacations, hiring, or firing that cause shifts originally assigned to go unstaffed. The new schedule will have to meet all the constraints of the original tour-schedule with the added goal of minimizing deviations. Minimizing deviations is an important goal of the new tour-schedule. The stability of the schedule is crucial for employee morale and the ability of employees to schedule their non-working lives.

The formulation and methodology presented in this article provide several contributions to the re-rostering literature. The following list summarizes the advancements proposed by this research.

- Provide a formulation that can account for heterogeneous workforce
- Insure adequate rest between shifts and limit the number of weekends assigned per month
- Describe a structured framework for obtaining employee shift preferences and managerial preferences for each employee
- Efficiently re-roster a tour-schedule while minimizing deviations

This article is organized in the following manner. Section 2 contains a survey of the pertinent literature and important previous work accomplished in nursing tour-scheduling. Section 3 details the re-rostering problem at Banner Health Baywood Hospital. Our methodology is general, but we feel that for realism and clarity we will introduce our case study early. Section 4 provides the proposed solution methodology. Section 5 shows the results of the methodology as applied to the case study. Section 6 finishes with conclusions and recommendations.

5.2 Literature Summary

A vast amount of scholastic work has been accomplished on the personnel scheduling problem. This work has been documented in a wide array of refereed journals, conference proceedings, and lecture notes. Personnel scheduling spans a wide range of problem instances, specific applications, solution techniques, and methodologies. Many survey papers of the pertinent

literature in the area have been written. Some of the most extensive and most recent include Ernst, et. al. (2004a), Ernst, et. al. (2004b), and Alfares (2004).

A primary application of the personal tour scheduling problem is found in “Solving the Problem of Re-rostering Nurse Schedules with Hard Constraints: New MultiCommodity Flow Models” by Margarida Moz and Margarida Vaz Pato. Published in the *Annals of Operations Research* in April 2004, the article proposes a method to re-roster a tour-schedule once changes in the given schedule no longer meet demand requirements. Sickness, termination, or hiring may be reasons for the need to change a given roster once it has been published.

Moz and Pato (2004) look at a continuous workforce that is heterogeneous. Differing nurse seniorities or specialties are taken into account by giving each shift in their model a separate task designation. The continuous workday is separated into 3 distinct shifts that start and end at the same time each day. These shifts are labeled Morning, Evening, and Night. A feasible schedule will meet hard constraints such as one shift per day, elimination of non-consecutive shifts, each shift/task combination assigned to a single nurse, the nurse assigned to a task must be eligible to perform the task, and a minimum number of days off per week for each nurse.

The algorithm approach proposed by Moz and Pato is an “integer multicommodity flow problem in a directed multi-level acyclic network” (2004). Each nurse is a commodity and is represented by an origin node in the network and by a destination node. The number of days to be scheduled is the number of

levels between the origin nodes and the destination node. Each non-origin/non-destination node in the network represents a task for a nurse to perform, including being off that day. The goal is to find a feasible path for each nurse through the network that meets all the hard constraints mentioned above. Below is a figure illustrating the network.

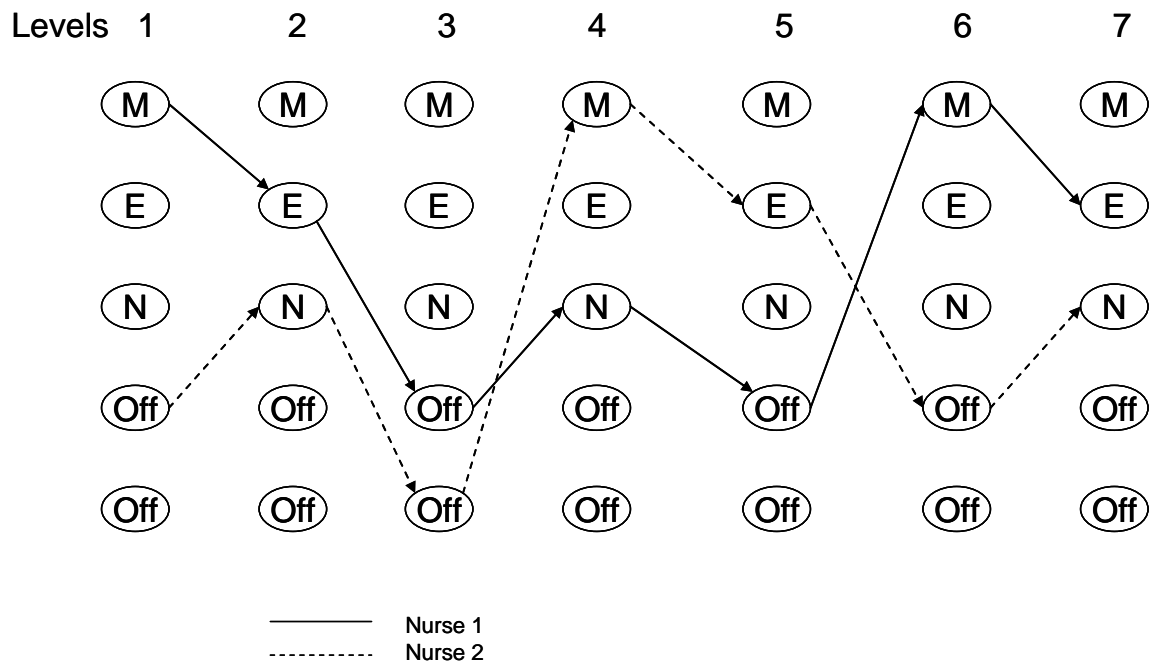


Figure 5.1: Multicommodity flow problem in a directed multi-level acyclic network

(Moz and Pato, 2004)

Moz and Pato use this network formulation to re-roster a completed 28 day schedule that has been perturbed at some day d , which is considered the origin of the network. “The re-rostering problem amounts to finding an optimal integer flow of n commodities, from the nodes of level $d-1$ to those of the last level” (Moz and Pato, 2004). The formulation is a mixed integer linear program

with binary constraints on the decision variables; they term this formulation as a binary linear program (BLP). The objective function of the algorithm is to find a new schedule that will minimize the number of nurses who must swap shifts.

The BLP does not use the network structure to solve the problem. The network structure is for illustrative purposes, as noted by Moz and Pato “this flow approach provides a suggestive network model for RSP (re-rostering problem), it leads to large scale BLP formulation, with many variables and constraints, some of which are additional to the flow model” (2004). Due to the low number of shifts, the formulation in effect enumerates all possible paths through a network that will satisfy the constraints described above. The nodes of the network represent shift/skill combinations and the arcs represent links between shifts that are feasible. Moz and Pato acknowledge this. “It is worth noting that... the flow of each commodity can only use one path from its origin to its destination, thus resulting in binary flows” (Moz and Pato, 2004). The commodities are the various nurses working a specific shift performing a given task. The origin is day d , while the destination is the 28th day.

The Moz and Pato models, as well as Jumard, et. al. (1998), Millar and Kiragu (1998), and Warner (1976), vary significantly from the research proposed by the network-based linear program formulation; although, both models use the network formulation for mostly illustrative purposes. The prime difference is the models presented by Moz and Pato are mixed integer linear programs with binary constraints. This requires a branch and bound approach to solve the problem. The network-based linear program requires no branching because the

solution to the linear program provides binary decision variables. The proposed network-based mathematical program (NBMP) can accommodate a large number of shifts with varied start times and of various lengths as well. The shifts in the NBMP research can also overlap. Moz and Pato describe a model where the shifts start and end at the same times each day and do not overlap. It is not readily apparent that their formulation could account for overlapping or varied length shifts. The enumeration of each feasible shift to shift to day off path through the network would make the problem grow rapidly if such variations in shift pattern were present. Next, although the workforce of nurses are heterogeneous in the aspect that they have varying skill sets, they do not have constraints on their availability apart from restrictions on consecutive shifts. The network-based mathematical program accounts for employee restrictions on availability as well as varying skill sets. Finally, the objective function of the Moz and Pato model is a deviation from a previously published tour-schedule. The objective of the proposed NBMP research is employee's weighted preference.

5.3 Re-rostering Case Study

The proposed re-rostering methodology will be demonstrated on an important real-world case study. The case study examined is a nurse scheduling problem instance at Banner Health Baywood Hospital in Phoenix, Arizona. The case study were chosen to illustrate the re-rostering approach and is presented without the loss of generality to other problem instances of personnel tour-scheduling. A brief description follows.

5.3.1 Banner Health Baywood Hospital

The second case study to re-roster is a nursing schedule at Banner Health Baywood Hospital. Banner Health is a large health care provider in Phoenix, Arizona and the surrounding areas. Banner runs many different hospitals that specialize in many different aspects of health care. One commonality from hospital to hospital is the need for tour-scheduling and then re-rostering its cadre of nurses. The cadre is made of many different specializations, different daily staffing levels, and hours of operation.

We relied on the expertise of James Broyles, an industrial engineer at one of Banner Health hospitals, to help us frame the problem as realistically as possible. Actual nurse preferences were not obtained; however, the production questionnaires or surveys that would have been needed to acquire such preferences are not important to the illustration of the methodology. Representative preferences were generated using the framework proposed in Section 5.4.

The Intensive Care Unit at Banner Health Mesa is staffed with qualified nurses 24 hours a day due to the nature of care required in a critical care situation. The number of nurses required each day varies with day of the week and month of the year. We will pick a representative number of 3 distinct shifts per day. Each distinct shift will be staffed by multiple nurses. The shifts will have staggering lunch hours and break times to insure adequate staffing. We will re-roster a cadre 40 nurses which will illustrate both a heavy workload and develop a new weekly schedule. Each nurse will not work more than 48 hours per week

or fewer than 24. This will represent the part-time and full-time mixture of nurses at Banner Health Baywood.

5.4 Proposed Tour-Schedule Solution Method and Model

The tour-scheduling problem is inherently a binary set-covering problem. Therefore, as problem size increases, the efficiency of the associated integer program decreases rapidly. However, we have developed a formulation of the problem as a minimum cost network-flow, using an arc capacity method, so that the resulting network structure generally provides integer binary answers. The network structure can easily be written as a linear program and solved using fast solution algorithms such as the CPLEX interior point method. Many solutions of this formulation do not require any branching, bounding, or cutting schemes to find integer solutions – the integer solutions are a consequence of the formulation method. The result being, very large problem instances can be solved in a computationally insignificant amount of time compared to the corresponding integer programming solution.

The model is detailed in Chapters 3 and 4. This model is named the network-based mathematical program (NBMP). The constraint set of this model requires all employees are scheduled only when available, have adequate skill level, meet minimum and maximum shifts per week, and have adequate rest between shifts. This constraint set will not change when re-rostering an optimal tour-schedule. The model's objective function will provide the mechanism for re-rostering and is presented below.

5.4.1 The Objective Function - Employee and Manager Preferences

The objective function of the tour-scheduling formulation, NBMP, is very important in the ability of the model to efficiently re-roster an optimal schedule. The objective function is made up of the preferences for each shift from the employees and by the ranking of the employees from the manager. It should be noted that there are two distinct situations when the formulation presented may cause the solver to find an alternate optimal solution that contains fractional employee assignments (there is always an integer solution). The first situation can be subverted through the intelligent use of preferences. This is discussed next and the reader is referred to the article in Chapter 4 for a discussion of the second.

Fractional solutions may occur when the preferences of a set of employees for a particular shift are equal and none is violating a consecutive shift restriction, maximum number of shifts per week constraint, or have more than the allotted weekend shifts. The solution is obviously an alternate optima, since the entire shift can be given to any of the fractional employees and the objective function will not change, nor is any constraint violated. This phenomenon can be combated by using relative rankings for both the shifts and the employees.

Relative rankings will break ties between the shift preferences of employees. Each employee, j , ranks all the shifts and each shift is given a score, $emppref_{i,j}$, on a relative scale, 1 through the number of shifts, ensuring no employee has the exact same preference for each shift, i . Next, the manager ranks all employees and then each employee is given the appropriate relative

score, $manpref_i$, ranging from 1 through number of employees. This number is then given an order of magnitude increase, O , over the scale for employee shift preferences and the two preferences are added together. The preferences of each shift/employee combination are the aggregate of the employee's and manager's preferences and are distinct from one another.

$$pref(s_{j,k,d}e_i) = emppref_{i,j} + (O \bullet manprep_i) \quad (4.8)$$

The objective function for the constraint set described in equations 4.1 – 4.6 then becomes

$$\text{MAX} \quad \sum_{j,k,d,i} s_{j,k,d} e_i \bullet pref(s_{j,k,d}e_i) \quad (4.9)$$

Figure 5.2 graphically illustrates the employee/shift preferences for the first eight shifts and ten employees after applying the above approach. The most important employee, as ranked by the manager, is on the top of the graph.

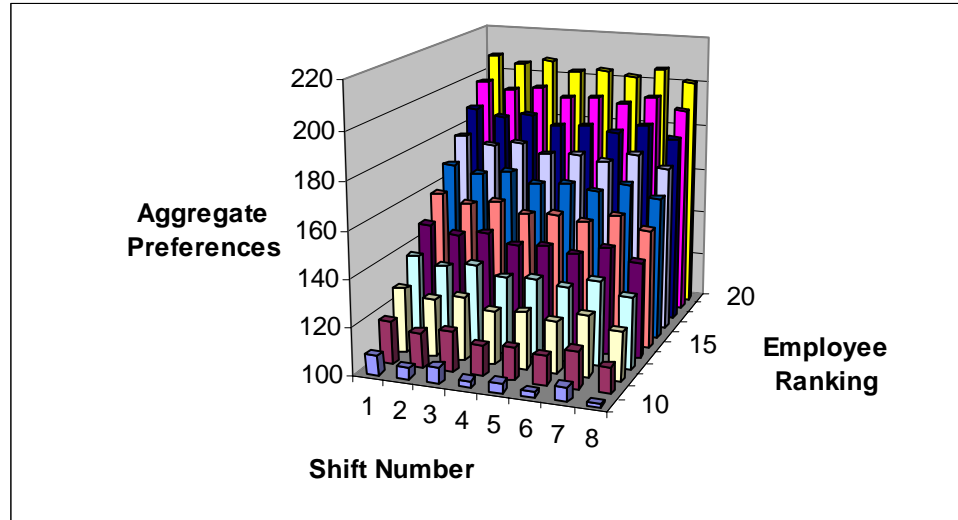


Figure 5.2: Illustration of aggregate preferences for employ/shift combination

This set of preferences is then modified to re-roster an optimal schedule. The goal of the re-rostered schedule is to have a minimal amount of deviations to the original schedule while covering the shift that is no longer staffed. The methodology is illustrated on a representative tour-scheduling problem instance. Figure 5.3 depicts the set of preferences selected in the original optimal schedule.

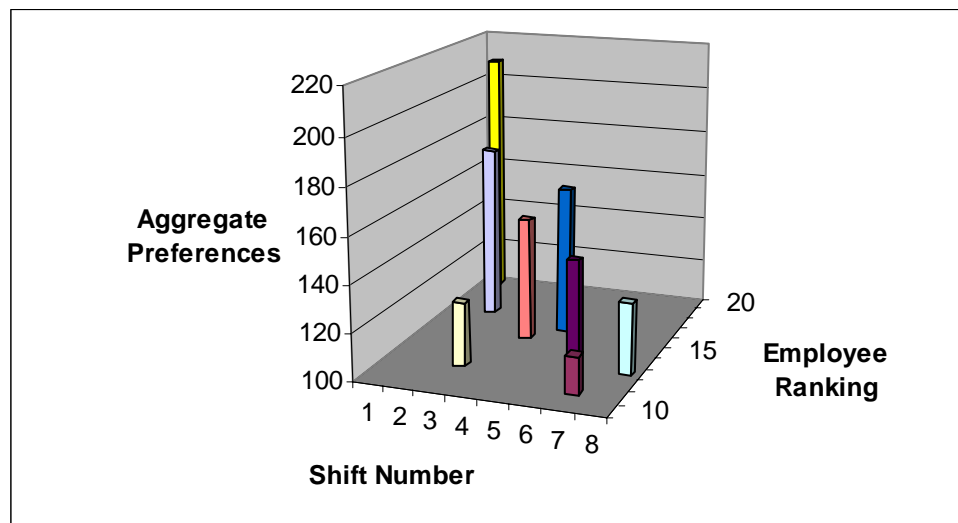


Figure 5.3: Preferences selected in original optimal schedule

The optimal schedule required all 8 shifts be staffed by only one of the 10 employees. These 8 employee/shift combinations constitute an optimal tour-schedule for a problem instance. Figure 5.4 then shows a perturbation in the schedule due to an employee's absence. In this illustration, employee 12 is unable to staff the shift for which they have been scheduled.

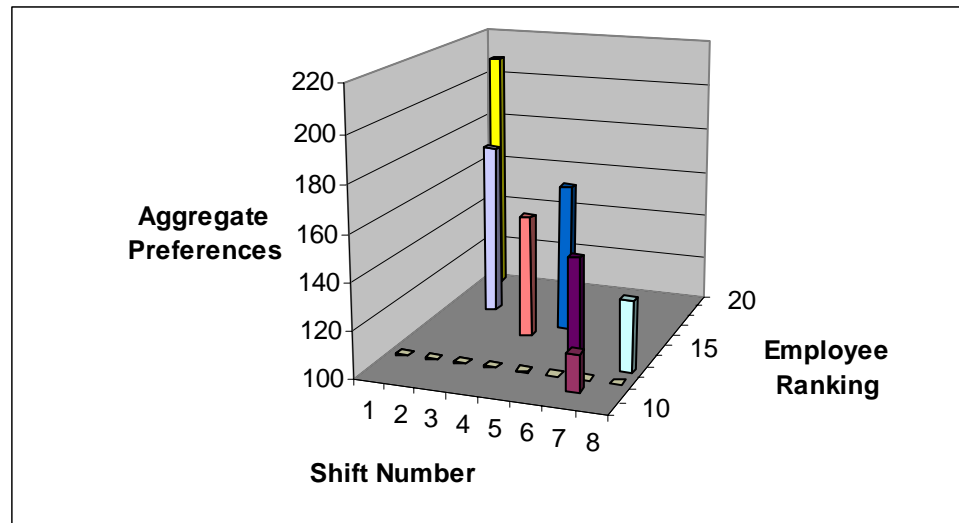


Figure 5.4: Absence of employee 12

We now apply our re-rostering methodology. In order to re-roster with minimal deviations to the original schedule, we modify the set of original preferences. Each preference for an employee who was assigned in the original schedule is increased by a factor M . The magnitude of factor M is problem dependent; however, a factor of three to five has worked well. The original preference set is then reentered into the model with the exception of employee twelve's preferences and the modified preferences of the currently scheduled employees. Figure 5.5 illustrates the modified preference set that will now be resolved to find a new schedule that will meet the shift requirement vacated by employee twelve while minimizing the changes to the original schedule.

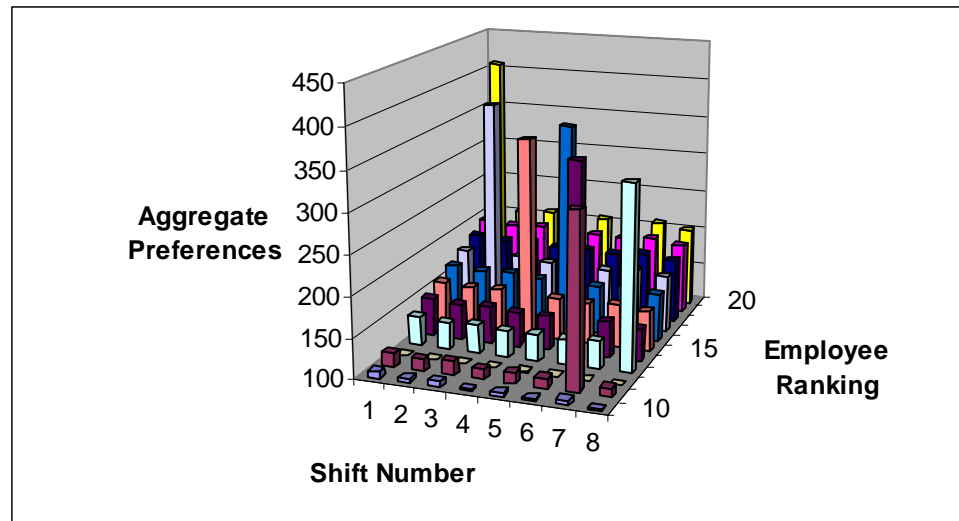


Figure 5.5: Modified preference set used in re-rostering

This methodology is now applied to the case study described above.

5.5 Re-rostering results

The re-rostering methodology shown in Section 5.4 is applied to the Banner Health case study. The results are presented below. The case study was re-rostered three times, using differing employee absentee scenarios. The first scenario is a relatively benign case of a single employee being absent from a single shift. The second scenario is more difficult, one employee absent for an entire week. Finally, the model will re-roster a schedule when five employees are absent for the entire week. The size of the problem instance for each case study, the number of deviations for each scenario, and the computational time to re-roster is shown in Table 5.1.

| Case Study | Number of Employees | Number of Shifts | Scenario | Number of Deviations | Computational Time (s) |
|---------------|------------------------|---------------------|-----------------|-------------------------|---------------------------|
| Banner Health | 20 | 63 | Single Shift | 0 | 0.01 |
| | | | Single Employee | 2 | 0.01 |
| | | | Five Employees | 3 | 0.03 |
| Banner Health | 40 | 126 | Single Shift | 0 | 0.04 |
| | | | Single Employee | 2 | 0.04 |
| | | | Five Employees | 10 | 0.06 |

Table 5.1: Re-rostering results

The results are very encouraging. Each scenario is re-rostered in a minimal amount of computational time and the deviations from the original schedule are minimal. The methodology works well for both case studies and finds a new optimal schedule for each instance.

5.6 Conclusion

The article presented a unique methodology to the very real problem of re-rostering a personnel tour-schedule. Employee schedules are inevitably in need of modification due to employee absenteeism. The chore of re-rostering a feasible schedule is not an easy one for a manager especially with the goal of minimizing the disturbances to the rest of the original schedule. The methodology here uses already obtained employee preferences to efficiently and with minimum deviations re-roster a published schedule. Minimizing deviations is an important goal of the new tour-schedule. The stability of the schedule is crucial for employee morale and the ability of employees to schedule their non-

working lives. Appendices E and F contain the Visual Basic Code used to formulate to modify the preferences for re-rostering and the CPLEX output respectively.

Chapter 6

CONTRIBUTIONS AND FUTURE RESEARCH

6.1 Contributions

This dissertation concentrated on developing a new representation of the personnel tour-scheduling problem. Personnel scheduling is one of the most difficult, important, and often studied problems in operations research. The optimal choice of the number of employees required to meet customer demand, shift start and stop times, daily lunches and breaks, and the assignment of the adequately skilled employee to the best shift is a large combinatorial optimization problem. Managers who attempt to manually solve the personnel scheduling problem expend many valuable work hours to find even a feasible solution which has little probability of being optimal based on any objective function.

Tour-scheduling, also known as the rostering problem in the literature, this dissertation presents a network-based mathematical programming model that can solve the tour-scheduling problem optimally using completely linear programming algorithms in many cases. In these cases, there is no need for branching, bounding, or cutting techniques, resulting in a savings of valuable computational time. In the cases where integer solutions are not found by the solver, the dissertation shows that the model produces solutions with a very small amount of fractional variables. Further, the fractional variables that are found are quickly converted to integer solutions via highly proven techniques such as Gomory fractional cuts.

The next fundamental contribution of this dissertation is the ability of the formulation to account for industries that have continuous (24-hour) operations. Modeling the rest periods between shifts during continuous operations has long been a limitation of tour-scheduling models. The network structure of the formulation proposed in this dissertation allows for specialized side constraints that restrict consecutive shifts for which an employee can be assigned. It is shown that these side constraints, in many cases, do not destroy the ability of the model to produce integer answers.

The ability to model heterogeneous workforces is another contribution of this dissertation. This collection of personnel who have significantly variant availabilities, skill sets, and wage rates is a very realistic makeup of today's workforces. Tour-scheduling a heterogeneous workforce must consider all of these differences and optimally match the personnel with the best shift possible and for which they are eligible. The model in this dissertation can be easily adapted to incorporate all the factors that are inherent in a heterogeneous workforce.

The formulation further allows for each employee to designate a preference for each shift in the schedule and for managers to rank the employees based on seniority or workplace performance. A tour-schedule that considers employee shift preferences and management employee weighting is very advantageous to an organization. If employees are matched to shifts that fit well with their home life, availabilities, work habits, or other personal considerations, they will be motivated to work harder and more efficiently. A

framework to collect and represent these preferences that is repeatable and comprehensible to the workforce is presented.

The formulation is then modified to account for two very important aspects of realistic employee rostering. The original model is stacked upon itself to allow for the management to schedule beyond the traditional weekly time horizon. The dissertation presents various multi-weekly models, including monthly, quarterly, and yearly. An additional network arc and node structure is then added to the model to insure that organization whose work week include all 7 days do not schedule employees for too many weekend shifts.

Finally, the model is once again modified to account for the inevitable perturbations that will occur in a tour-schedule, especially those of a multi-week length. These fluctuations will be the result of employee sicknesses, vacations, hiring, or firing. This dissertation demonstrates how the model can be used to re-roster a previous optimal schedule. The modification is powerful, but causes little complication in the way of reformulation or expansion of the computational time of the solver.

A timely and optimal solution to the personnel scheduling problem will save each manager valuable time, meet customer demand, and increase worker morale by matching each employee to the best shift possible. This dissertation presents an extremely fast, computationally efficient, and optimal network-flow based mathematical programming solution for the extremely challenging problem of rostering and re-rostering a continuous heterogeneous workforce.

6.2 Future Research

There are many areas of potential for future research. Three of these areas involve refining the current model, applying the network-based model to a different scheduling problem, developing robust tour-schedules against employee absenteeism, and combining the shift selection and tour-scheduling problems.

The current model can be refined by defining a set of rules for the preferences in the objective function. A real advancement would be a set of rules that gave the model the best chance to find an integer solution without the need for an integer constraint set. The set of rules must incorporate the number of shifts and employees, and the rest required between consecutive shifts.

The model is flexible enough that it could be applied to another scheduling problem. Academic scheduling is a classic operations research problem that might be solved using the model and formulation proposed here. The model would have to be modified to account for room scheduling in addition to the students and faculty. However, many of the same constraints associated with the tour-scheduling model would also be found in the academic scheduling problem. The model proposed in this dissertation could efficiently handle the academic scheduling constraints just as well.

Employee absenteeism is an unavoidable consequence of employee tour-scheduling. A modification of the re-rostering objective function to find schedules that may not be optimal in terms of preferences, but are more robust to absent employees would be a valuable extension to the model. Furthermore, a design of experiments approach to quantifying the size of the Big M reward to the

objective function for optimally scheduled employees would provide the manager with important information regarding the number of deviations that may be tolerated.

Finally, using a network-based model to solve the combined shift selection and tour-scheduling problem would be a powerful tool. The model would have to be modified considerably to account for the non-linearity aspects of the combined model. A network-based model with an accompanying heuristic may solve this problem. A more detailed of this extension is discussion is presented next.

6.2.1 Combined Shift Selection/Tour-Scheduling

Optimally solving the tour scheduling sub-problem does not guarantee a global optimal solution to the combined shift selection/tour scheduling problem. An alternate set of shifts that minimizes excess employee hours but increases the number of preferred shifts of personnel can only be found by solving the combined shift selection/tour-scheduling problem. The advantage of a global optimal, and the reason to pursue a solution, is the possibility of bettering the tour-scheduling preferences while still minimizing the number of excess hours devoted to shifts covering the required demand.

The proposed solution method presented in this dissertation was expanded to solve the combined problem. However, a pure network-flow structure is problematic for the combined problem. In short, the balance equations associated with the network-flow representation of the shift selection/tour-scheduling problem are non-linear. Figure 6.1 shows an illustration of the network framework for the combined problem. The extreme

right hand side of the figure is not shown, but will look identical to the network representations presented earlier.

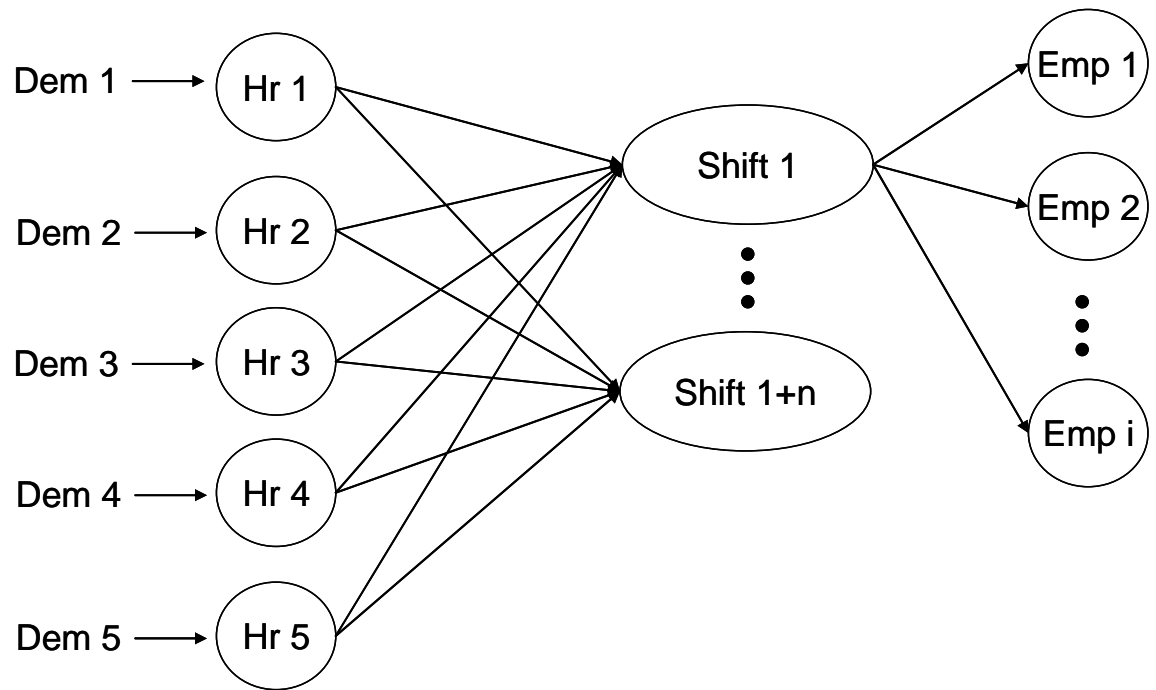


Figure 6.1: Network-flow representation of combined problem

Here, the nodes HR represent each individual hour of the day. The increment could easily be increase to multiple hours, or decrease to fractions of an hour. The increment should correspond to the profile produced by the forecasting models during the demand modeling sub-problem. The demand flows from each hour to a corresponding shift whose start and stop times encompass the particular hourly node. The shift is either then selected to fulfill the demand associated with that hour of the day, or a different shift encompassing that hour is selected. Once a shift is selected, the model

executes just as described in this dissertation. However, this network representation of the combined problem leads to a constraint set that is non-linear.

The non-linearity is obvious when the balance equations for examples from the hourly and shift nodes are written. Equation 6.1 is the balance equation for node HR1. Theses equations are representative of all the balance equations for all the hourly nodes.

$$Hr1s1 + Hr1s2 + \dots + Hr1sj = Dem1 \quad (6.1)$$

Equation 6.1 is perfectly linear, however, the balance equations for the shift nodes are not. Equation 6.2 shows the non-linearity. The RHS must be divided by the number of hourly nodes flowing into the shift node, in order for the rest of the network to schedule only one shift. Without the normalization, the network would attempt to schedule the shift as many times as hourly demand was met with that shift.

$$Hr1s1 + Hr2s1 + \dots + Hr5s1 = \left[\frac{1}{Hr1s1 + Hr2s1 + \dots + Hr5s1} \right] \bullet (s1e1 + s1e2 + \dots + s1e_i)$$

This is the simplest case. A more complicated case involving multiple demands for an hour (a likely case) or varying shift lengths make the problem even more difficult. Figure 6.2 shows a more realistic flow diagram for the first half of the network representation of the combined problem.

The non-linearity requires different solution techniques than those presented in this dissertation. Conjugate gradient methods or heuristics may be a viable option for solving the combined problem. Another solution approach may be an iterative method. The approach would solve the two sub-problems once again separately, but return to each sub-problem with information from the solution of the other that can better the global optima.

REFERENCES

- Alfares, H. (2004). Survey, categorization, and comparison of recent tour scheduling literature. *Annals of Operations Research, Special Issue on Staff Scheduling and Rostering*, 127, 145-175.
- Ashley, D.W. (1995). Spreadsheet optimization system for library staff scheduling. *Computers and Operations Research*, 22(6), 615–624.
- Ball, M. & Benoit-Thompson, H. (1988). “A lagrangian relaxation based heuristic for the urban transit crew scheduling problem. In J. Daduna and I. Wren (Eds.). *Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, 308, 54–67.
- Banihashemi, M. & Haghani, A. (2001). A new model for the mass transit crew scheduling problem. In S. Voss and J. Daduna (Eds.). *Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems*, 505, 1–16.
- Barnhart, C., E. Johnson, R. Anbil, & Hatay, L. (1994). A column generation technique for the long-haul crew assignment problem. In T. Ciriani and R. Leachman (Eds.), *Optimization in Industry*, 2, 7–24. Chichester: Wiley.
- Borndörfer, R., M. Grötschel, & Löbel, A. (2001). Duty scheduling in public transit. *Technical Report, ZIB*.
- Brusco, M. & Johns, T. (1995). Improving the dispersion of surplus labor in personnel scheduling solutions. *Computers and Industrial Engineering*, 28(4), 745-754.
- Carraraesi, P. & Gallo, G. (1984b). Network models for vehicle and crew scheduling. *European Journal of Operational Research*, 16(2), 139–151.
- Dowsland, K. & Thompson, J. (2000). Solving a nurse scheduling problem with knapsacks, networks and tabu Search. *Journal of the Operational Research Society*, 51, 825–833.
- Ernst, A., H. Jiang, M. Krishnamoorthy, & Sier, D. (2004a). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153, 3-27.
- Ernst, A., H. Jiang, M. Krishnamoorthy, B. Owens, & Sier, D. (2004b). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research, Special Issue on Staff Scheduling and Rostering*, 127, 21-141.

- Eveborn, P. & Ronnqvist, M. (2004). Scheduler – a system for staff planning. *Annals of Operations Research, Special Issue on Staff Scheduling and Rostering*, 128, 21–45.
- Glover, F. & McMillan, C. (1986). The general employee scheduling problem: an integration of MS and AI. *Computers and Operations Research*, 13(5), 563–573.
- Glover, F., C. McMillan, & Grover, R. (1985). A heuristic programming approach to the employee scheduling problem and some thoughts on ‘managerial robots’. *Journal of Operations Management*, 4, 113–128.
- Haase, K. (1999). Advanced column generation techniques with applications to marketing, retail and logistics management. *Ph.D. Thesis, Habilitation Thesis, University of Kiel*.
- Jiang, H., M. Krishnamoorthy, & Sier, D. (2004). Preface: special issue: staff scheduling and rostering: theory and applications, part I. *Annals of Operations Research, Special Issue on Staff Scheduling and Rostering*, 127, 17-20.
- Jaumard, B., F. Semet, & Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European Journal of Operational Research*, 107(1), 1–18.
- Lauer, J., L.W. Jacobs, M.J. Brusco, & Bechtold, S.E. (1994). An interactive optimization-based decision support system for scheduling part-time computer lab attendants. *Omega*, 22(6), 613–626.
- Li, C., E. Robinson, & Mabert, V. (1991). An evaluation of tour scheduling heuristics with differences in employee productivity and cost. *Decision Sciences*, 22, 700–718.
- Litchfield, J., A. Ingolfsson, & Cheng, K. (2003). Rostering for a restaurant. *INFOR*, 41(3), 287-300.
- Loucks, J. & Jacobs, F. (1991). Tour scheduling and task assignment of a heterogeneous work force: a heuristic approach. *Decision Sciences*, 22(4), 719–739.
- Love, R. & Hoey, J. (1990). Management science improves fast food operations. *Interfaces*, 20(2), 21–29.
- Mabert, V. & Raedels, A. (1977). The detail scheduling of a part-time work force: a case study of teller staffing. *Decision Sciences*, 8, 109–120.

- Mabert, V. & Watts, C. (1982). A simulation analysis of tour-shift construction procedures. *Management Science*, 28(5), 520–532.
- Mabert, V. & Showalter, M. (1990). Measuring the impact of part-time workers in service organizations. *Journal of Operations Management*, 9(2), 209–229.
- Mason, A. & Smith, M. (1998). A nested column generator for solving rostering problems with integer programming. In L. Caccetta, K. Teo, P. Siew, Y. Leung, L. Jennings, and V. Rehbock (Eds.), *International Conference on Optimization: Techniques and Applications*.
- Mellouli, T. (2001). A network flow approach to crew scheduling based on an analogy to a vehicle maintenance routing problem. In S. Voss and J. Daduna (eds.), *Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems*, 505, 91–120. Springer.
- Millar, H. & Kiragu, M. (1998). Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. *European Journal of Operational Research*, 104(3), 582–592.
- Mould, G. (1996). Case study of manpower planning for clerical operations. *Journal of the Operational Research Society*, 47(3), 358–368.
- Moz, M. & Pato, M.V. (2004). Solving the problem of rerostering nurse schedules with hard constraints: new multicommodity flow models. *Annals of Operations Research, Special Issue on Staff Scheduling and Rostering*, 128, 179–197.
- Nicoletti, B. (1975). Automatic crew rostering. *Transportation Science* 9(1), 33–42.
- Patrikalakis, I. & Xerocostas, D. (1992). A new decomposition scheme of the urban public transport scheduling problem. In M. Desrochers and J. Rousseau (eds.), *Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, 386, 407–425.
- Thompson, G. (1996b). A simulated-annealing heuristic for shift scheduling using non-continuously available employees. *Computers and Operations Research* 23(3), 275–288.
- Tingley, G. (1979). Still another solution method for the monthly aircrew assignment problem. In *1979 AGIFORS Symposium Proceedings*, 19, 143–203.

Warner, M.D. (1976). Scheduling nursing personnel according to nursing preference: a mathematical programming approach. *Operations Research*, 24(5), 842-856.

Yan, S. & Chang, J. (2002). Airline cockpit crew scheduling. *European Journal of Operational Research*, 136(3), 501–511.

APPENDIX A

CHAPTER THREE VISUAL BASIC CODE

Private Sub WriteNetLP()

'This subroutine will write out in CPLEX .LP format
 'a LP for solving large crew scheduling problems
 'using a network-based linear program
 'Shane Knighton 6-20-04

'Rev 2 9-22-04 - added Shift Demand Matrix allowing for
 'variable number of employees needed per shift

'Dimensionalize variables
 Dim NumShiftsM, NumShiftsT, NumShiftsW, NumShiftsR, NumShiftsF,
 NumShiftsS, NumShiftsN As Integer
 Dim NumShifts, dayShifts, NumDays, NumEmp As Integer
 Dim MaxNumShifts, MinNumShifts, ShiftDemand As Integer
 Dim i, j, k As Integer
 Dim Name, filepath, text2write As String
 Dim varName As String

'ConflictMatrix is a NumShift by maximum number of conflicts
 'matrix that has a 1 in the i,j column if shift i and j
 'are conflicted
 Dim ConflictMatrix() As Integer
 Dim MaxNumConfl As Integer

'PriorityMatrix is a shift by employee matrix
 'containing the priority of each shift for each employee
 Dim PriorityMatrix() As Single

'ShiftDemMatrix is the number of employees needed for each shift
 Dim ShiftDemMatrix() As Integer

'Problem size data
 NumDays = 7
 NumEmp = 50
 'Need to get the number of shifts on each day
 NumShiftsM = 21
 NumShiftsT = 34
 NumShiftsW = 34
 NumShiftsR = 34
 NumShiftsF = 35
 NumShiftsS = 24
 NumShiftsN = 13
 'Total all the shifts

```

NumShifts = NumShiftsM + NumShiftsT + NumShiftsW + NumShiftsR +
NumShiftsF + NumShiftsS + NumShiftsN
'Min and Max shifts per week per employee
MaxNumShifts = 5
MinNumShifts = 0 '2
'Maximum number of conflicted shifts
MaxNumConfl = 30

'Get Name of Problem
Name = "run5ahp" 'ActiveSheet.Name

'Size ConfliftMatrix and ConflictPointer
ReDim ConflictMatrix(1 To NumShifts, 1 To MaxNumConfl) As Integer

'Populate ConflictMatrix
Dim startRow, startCol As Integer
startRow = 10 + NumShifts
startCol = 3

For i = 1 To NumShifts
    For j = 1 To MaxNumConfl
        If Cells(startRow + i, startCol + j) <> 0 Then
            ConflictMatrix(i, j) = Cells(startRow + i, startCol + j)
        End If
    Next j
Next i

'Size PriorityMatrix
ReDim PriorityMatrix(1 To NumShifts, 1 To NumEmp) As Single

'Populate PriorityMatirix
startRow = 5
startCol = 6

For i = 1 To NumShifts
    For j = 1 To NumEmp
        PriorityMatrix(i, j) = Cells(startRow + i, startCol + j) ' + Round(Rnd(), 2)
    Next j
Next i

'Size ShiftDemMatrix
ReDim ShiftDemMatrix(1 To NumShifts) As Integer

'Populate ShiftDemMatrix
startRow = 5

```



```
startCol = 5
```

```
For i = 1 To NumShifts
```

```
    ShiftDemMatrix(i) = Cells(startRow + i, startCol)
```

```
Next i
```

```
'Open a file to write to
```

```
filepath = "C:\Documents and Settings\Shane\My Documents\Dissertation\Phase  
1\" & Name & ".lp"
```

```
Open filepath For Output As #1
```

```
'Write header and objective function
```

```
Print #1, "MAXIMIZE"
```

```
Print #1, "Z:"
```

```
'i is the loop for shifts
```

```
For i = 1 To NumShifts
```

```
    'j is the loop for employees
```

```
    For j = 1 To NumEmp
```

```
        'Concatenate variable name
```

```
        varName = "s" & i & "e" & j
```

```
        'Write variables and cost coefficients to file
```

```
        'Write final variable and cost to file, leave off the "+"
```

```
        If i = NumShifts And j = NumEmp Then
```

```
            Print #1, PriorityMatrix(i, j) & " " & varName
```

```
        Else
```

```
            Print #1, PriorityMatrix(i, j) & " " & varName & " +"
```

```
        End If
```

```
    Next j
```

```
Next i
```

```
'Write constraints
```

```
'Header
```

```
Print #1, "SUBJECT TO"
```

```
'Constraint number counter
```

```
Dim ConstNum As Double
```

```
ConstNum = 1
```

```
'Define the 6 sets of constraints
```

```
'Set1: shifts = demand equality constraints
```

```
'    s001e01 + ... + s001e50 = demand
```

```
'Set2: total shifts per employee for each day
```

```
'    will be required to be less than one
```

```
'    by set 5, needed to total number
```

```
'    of shifts per week
```

```

'      s001e01 + ... + s025e01 = me01
'Set3: each employee has less than max number of shifts per week
'      me01 + ... + ne01 <= max number of shifts
'Set4: each employee has more than min number of shifts per week
'      me01 + ... + ne01 >= min number of shifts
'Set5: only one shift per day per employee
'      me01 <= 1
'Set6: shift confliction for each employee
'      example if shifts 25 and 26 are conflicted
'      s025e01 + s026e01 <= 1
'Write Set1 constraints
'i is the loop for shifts
For i = 1 To NumShifts
    'Write constraint lable
    Print #1, "c" & ConstNum & ":"
    'j is the loop for employees
    For j = 1 To NumEmp
        'Concatenate variable name
        varName = "s" & i & "e" & j
        'Write variables and coefficients to file, leave off "+" for final emp
        If j = NumEmp Then
            Print #1, varName & " = " & ShiftDemMatrix(i)
        Else
            Print #1, varName & " +"
        End If
    Next j
    ConstNum = ConstNum + 1
Next i

'Write Set2 constraints
'i is the loop for shifts
i = 1
Do While i < NumShifts

    'Determine what day this shift is on
    If i < NumShiftsM Then dayNum = 1
    If i > NumShiftsM Then dayNum = 2
    If i > NumShiftsM + NumShiftsT Then dayNum = 3
    If i > NumShiftsM + NumShiftsT + NumShiftsW Then dayNum = 4
    If i > NumShiftsM + NumShiftsT + NumShiftsW + NumShiftsR Then dayNum =
5
    If i > NumShiftsM + NumShiftsT + NumShiftsW + NumShiftsR + NumShiftsF
Then dayNum = 6
    If i > NumShiftsM + NumShiftsT + NumShiftsW + NumShiftsR + NumShiftsF +
NumShiftsS Then dayNum = 7

```

```

'Find the right day
If dayNum = 1 Then
    dayName = "m"
    dayShifts = NumShiftsM
Elseif dayNum = 2 Then
    dayName = "t"
    dayShifts = NumShiftsT
Elseif dayNum = 3 Then
    dayName = "w"
    dayShifts = NumShiftsW
Elseif dayNum = 4 Then
    dayName = "r"
    dayShifts = NumShiftsR
Elseif dayNum = 5 Then
    dayName = "f"
    dayShifts = NumShiftsF
Elseif dayNum = 6 Then
    dayName = "s"
    dayShifts = NumShiftsS
Elseif dayNum = 7 Then
    dayName = "n"
    dayShifts = NumShiftsN
Else
    GoTo ErrorHandler
End If

'j is the loop for employees
For j = 1 To NumEmp
    'Set2 constraints
    Print #1, "c" & ConstNum & ":"
    For k = i To i - 1 + dayShifts
        'Concatenate variable name
        varName = "s" & k & "e" & j
        'Write variables and coefficients to file
        If k = i - 1 + dayShifts Then
            Print #1, varName & " - " & dayName & j & " = 0"
        Else
            Print #1, varName & " + "
        End If
    Next k
    ConstNum = ConstNum + 1
Next j
i = i + dayShifts
Loop

```

'Write Set3 constraints

For j = 1 To NumEmp

 Print #1, "c" & ConstNum & ":"

 Print #1, "m" & j & " + t" & j & " + w" & j & " + r" & j & " + f" & j & " + s" & j & " + n" & j & " <= " & MaxNumShifts

 ConstNum = ConstNum + 1

Next j

'Write Set4 constraints

For j = 1 To NumEmp

 Print #1, "c" & ConstNum & ":"

 Print #1, "m" & j & " + t" & j & " + w" & j & " + r" & j & " + f" & j & " + s" & j & " + n" & j & " >= " & MinNumShifts

 ConstNum = ConstNum + 1

Next j

'Write Set5 constraints

For j = 1 To NumEmp

 Print #1, "c" & ConstNum & ":"

 Print #1, "m" & j & " <= 1"

 ConstNum = ConstNum + 1

 Print #1, "c" & ConstNum & ":"

 Print #1, "t" & j & " <= 1"

 ConstNum = ConstNum + 1

 Print #1, "c" & ConstNum & ":"

 Print #1, "w" & j & " <= 1"

 ConstNum = ConstNum + 1

 Print #1, "c" & ConstNum & ":"

 Print #1, "r" & j & " <= 1"

 ConstNum = ConstNum + 1

 Print #1, "c" & ConstNum & ":"

 Print #1, "f" & j & " <= 1"

 ConstNum = ConstNum + 1

 Print #1, "c" & ConstNum & ":"

 Print #1, "s" & j & " <= 1"

 ConstNum = ConstNum + 1

 Print #1, "c" & ConstNum & ":"

 Print #1, "n" & j & " <= 1"

 ConstNum = ConstNum + 1

Next j

'Write Set6 constraints

For i = 1 To NumShifts

 For k = 1 To MaxNumConfl

```

    If ConflictMatrix(i, k) <> 0 Then
        For j = 1 To NumEmp
            Print #1, "c" & ConstNum & ":"
            Print #1, "s" & i & "e" & j & " + s" & ConflictMatrix(i, k) & "e" & j & " <= 1"
            ConstNum = ConstNum + 1
        Next j
    End If
Next k
Next i

```

```

'Write Integer requirements
'Print #1, "INTEGER"
'For i = 1 To NumShifts
'    For j = 1 To NumEmp
'        Print #1, "s" & i & "e" & j
'    Next j
'Next i

```

```

'Write END
Print #1, "END"
'Close mps file
Close #1

```

```
Exit Sub
```

```

ErrorHandler:
MsgBox ("Error")
'Close mps file
Close #1

```

```
End Sub
```

```
Private Sub WriteConflicts()
```

```

    rest = 12
    totshifts = 195

    For i = 1 To totshifts
        curshift = Cells(i + 5, 2).Value
        dy = Int(curshift / 144) + 1
        begin = Int(curshift / 6)
        dur = curshift - (begin * 6) + 2
        finish = begin + dur

        endrest = finish + rest
    
```

```

    k = 4
    For j = 1 To totshifts
        compareshift = Cells(j + 5, 2).Value
        begin2 = Int(compareshift / 6)
        If compareshift > dy * 144 And begin2 < endrest Then
            Cells(i + 10 + totshifts, k).Value = Cells(j + 5, 3).Value
            k = k + 1
        End If
    Next j
Next i

End Sub

Private Sub Priorities()

End Sub

startRow = 5
startCol = 76

For i = 1 To 285
    shift = Cells(startRow + i, startCol).Value
    emp = Cells(startRow + i, startCol + 2).Value
    Cells(startRow + i, startCol + 5).Value = Cells(shift + 5, emp + 6).Value
Next i

Private Sub Values()

startRow = 5
startCol = 76

For i = 1 To 285
    shift = Cells(startRow + i, startCol).Value
    emp = Cells(startRow + i, startCol + 2).Value
    Cells(5 + shift, 87 + emp).Value = Cells(startRow + i, startCol + 4).Value
Next i

End Sub

```

APPENDIX B

CHAPTER THREE CPLEX OUTPUT

Log started (V8.1.0) Thu Oct 28 16:46:53 2004

Problem 'Runlnflp.lp' read.

Read time = 0.03 sec.

Tried aggregator 1 time.

LP Presolve eliminated 350 rows and 0 columns.

Reduced LP has 645 rows, 10100 columns, and 20550 nonzeros.

Presolve time = 0.02 sec.

Initializing dual steep norms . . .

Iteration log . . .

Iteration: 1 Dual objective = 38722.250000

Dual simplex - Optimal: Objective = 3.8625380000e+04

Solution time = 0.03 sec. Iterations = 125 (0)

| Variable Name | Solution Value |
|---------------|----------------|
| s1e20 | 1.000000 |
| s2e5 | 1.000000 |
| s3e6 | 1.000000 |
| s4e3 | 1.000000 |
| s5e50 | 1.000000 |
| s6e13 | 1.000000 |
| s7e25 | 1.000000 |
| s8e39 | 1.000000 |
| s9e10 | 1.000000 |
| s10e41 | 1.000000 |
| s11e49 | 1.000000 |
| s12e48 | 1.000000 |
| s13e35 | 1.000000 |
| s14e29 | 1.000000 |
| s15e11 | 1.000000 |
| s16e40 | 1.000000 |
| s17e28 | 1.000000 |
| s18e21 | 1.000000 |
| s19e19 | 1.000000 |
| s20e14 | 1.000000 |
| s21e47 | 1.000000 |
| s22e8 | 1.000000 |
| s23e32 | 1.000000 |
| s24e48 | 1.000000 |
| s25e2 | 1.000000 |
| s26e19 | 1.000000 |
| s27e50 | 1.000000 |
| s28e7 | 1.000000 |
| s29e13 | 1.000000 |
| s30e40 | 1.000000 |
| s31e20 | 1.000000 |
| s32e39 | 1.000000 |
| s33e49 | 1.000000 |
| s34e10 | 1.000000 |
| s35e21 | 1.000000 |
| s36e25 | 1.000000 |

| | |
|--------|----------|
| s37e6 | 1.000000 |
| s38e45 | 1.000000 |
| s39e17 | 1.000000 |
| s40e31 | 1.000000 |
| s41e18 | 1.000000 |
| s42e46 | 1.000000 |
| s43e42 | 1.000000 |
| s44e26 | 1.000000 |
| s45e27 | 1.000000 |
| s46e14 | 1.000000 |
| s47e47 | 1.000000 |
| s48e23 | 1.000000 |
| s49e16 | 1.000000 |
| s50e28 | 1.000000 |
| s51e4 | 1.000000 |
| s52e3 | 1.000000 |
| s53e11 | 1.000000 |
| s54e37 | 1.000000 |
| s55e34 | 1.000000 |
| s56e42 | 1.000000 |
| s57e30 | 1.000000 |
| s58e49 | 1.000000 |
| s59e19 | 1.000000 |
| s60e14 | 1.000000 |
| s61e18 | 1.000000 |
| s62e7 | 1.000000 |
| s63e41 | 1.000000 |
| s64e32 | 1.000000 |
| s65e46 | 1.000000 |
| s66e6 | 1.000000 |
| s67e1 | 1.000000 |
| s68e27 | 1.000000 |
| s69e36 | 1.000000 |
| s70e24 | 1.000000 |
| s71e17 | 1.000000 |
| s72e15 | 1.000000 |
| s73e44 | 1.000000 |
| s74e39 | 1.000000 |
| s75e16 | 1.000000 |
| s76e45 | 1.000000 |
| s77e8 | 1.000000 |
| s78e37 | 1.000000 |
| s79e9 | 1.000000 |
| s80e28 | 1.000000 |
| s81e33 | 1.000000 |
| s82e13 | 1.000000 |
| s83e40 | 1.000000 |
| s84e12 | 1.000000 |
| s85e10 | 1.000000 |
| s86e26 | 1.000000 |
| s87e25 | 1.000000 |
| s88e34 | 1.000000 |
| s89e43 | 1.000000 |
| s90e13 | 1.000000 |
| s91e39 | 1.000000 |

| | |
|---------|----------|
| s92e41 | 1.000000 |
| s93e19 | 1.000000 |
| s94e10 | 1.000000 |
| s95e40 | 1.000000 |
| s96e42 | 1.000000 |
| s97e45 | 1.000000 |
| s98e30 | 1.000000 |
| s99e36 | 1.000000 |
| s100e33 | 1.000000 |
| s101e23 | 1.000000 |
| s102e1 | 1.000000 |
| s103e38 | 1.000000 |
| s104e2 | 1.000000 |
| s105e26 | 1.000000 |
| s106e44 | 1.000000 |
| s107e28 | 1.000000 |
| s108e29 | 1.000000 |
| s109e35 | 1.000000 |
| s110e7 | 1.000000 |
| s111e47 | 1.000000 |
| s112e31 | 1.000000 |
| s113e46 | 1.000000 |
| s114e12 | 1.000000 |
| s115e21 | 1.000000 |
| s116e50 | 1.000000 |
| s117e9 | 1.000000 |
| s118e34 | 1.000000 |
| s119e15 | 1.000000 |
| s120e8 | 1.000000 |
| s121e27 | 1.000000 |
| s122e37 | 1.000000 |
| s123e22 | 1.000000 |
| s124e34 | 1.000000 |
| s125e44 | 1.000000 |
| s126e28 | 1.000000 |
| s127e31 | 1.000000 |
| s128e38 | 1.000000 |
| s129e35 | 1.000000 |
| s130e50 | 1.000000 |
| s131e30 | 1.000000 |
| s132e11 | 1.000000 |
| s133e9 | 1.000000 |
| s134e19 | 1.000000 |
| s135e4 | 1.000000 |
| s136e37 | 1.000000 |
| s137e8 | 1.000000 |
| s138e16 | 1.000000 |
| s139e41 | 1.000000 |
| s140e3 | 1.000000 |
| s141e1 | 1.000000 |
| s142e23 | 1.000000 |
| s143e48 | 1.000000 |
| s144e22 | 1.000000 |
| s145e21 | 1.000000 |
| s146e10 | 1.000000 |

| | |
|---------|----------|
| s147e14 | 1.000000 |
| s148e2 | 1.000000 |
| s149e18 | 1.000000 |
| s150e15 | 1.000000 |
| s151e36 | 1.000000 |
| s152e12 | 1.000000 |
| s153e24 | 1.000000 |
| s154e46 | 1.000000 |
| s155e13 | 1.000000 |
| s156e17 | 1.000000 |
| s157e47 | 1.000000 |
| s158e5 | 1.000000 |
| s159e15 | 1.000000 |
| s160e40 | 1.000000 |
| s161e5 | 1.000000 |
| s162e2 | 1.000000 |
| s163e34 | 1.000000 |
| s164e35 | 1.000000 |
| s165e9 | 1.000000 |
| s166e18 | 1.000000 |
| s167e11 | 1.000000 |
| s168e41 | 1.000000 |
| s169e32 | 1.000000 |
| s170e37 | 1.000000 |
| s171e43 | 1.000000 |
| s172e25 | 1.000000 |
| s173e7 | 1.000000 |
| s174e49 | 1.000000 |
| s175e50 | 1.000000 |
| s176e8 | 1.000000 |
| s177e31 | 1.000000 |
| s178e14 | 1.000000 |
| s179e3 | 1.000000 |
| s180e20 | 1.000000 |
| s181e27 | 1.000000 |
| s182e24 | 1.000000 |
| s183e33 | 1.000000 |
| s184e3 | 1.000000 |
| s185e9 | 1.000000 |
| s186e11 | 1.000000 |
| s187e15 | 1.000000 |
| s188e16 | 1.000000 |
| s189e17 | 1.000000 |
| s190e44 | 1.000000 |
| s191e39 | 1.000000 |
| s192e7 | 1.000000 |
| s193e26 | 1.000000 |
| s194e42 | 1.000000 |
| s195e49 | 1.000000 |
| m3 | 1.000000 |
| m5 | 1.000000 |
| m6 | 1.000000 |
| m10 | 1.000000 |
| m11 | 1.000000 |
| m13 | 1.000000 |

| | |
|-----|----------|
| m14 | 1.000000 |
| m19 | 1.000000 |
| m20 | 1.000000 |
| m21 | 1.000000 |
| m25 | 1.000000 |
| m28 | 1.000000 |
| m29 | 1.000000 |
| m35 | 1.000000 |
| m39 | 1.000000 |
| m40 | 1.000000 |
| m41 | 1.000000 |
| m47 | 1.000000 |
| m48 | 1.000000 |
| m49 | 1.000000 |
| m50 | 1.000000 |
| t2 | 1.000000 |
| t3 | 1.000000 |
| t4 | 1.000000 |
| t6 | 1.000000 |
| t7 | 1.000000 |
| t8 | 1.000000 |
| t10 | 1.000000 |
| t11 | 1.000000 |
| t13 | 1.000000 |
| t14 | 1.000000 |
| t16 | 1.000000 |
| t17 | 1.000000 |
| t18 | 1.000000 |
| t19 | 1.000000 |
| t20 | 1.000000 |
| t21 | 1.000000 |
| t23 | 1.000000 |
| t25 | 1.000000 |
| t26 | 1.000000 |
| t27 | 1.000000 |
| t28 | 1.000000 |
| t31 | 1.000000 |
| t32 | 1.000000 |
| t34 | 1.000000 |
| t37 | 1.000000 |
| t39 | 1.000000 |
| t40 | 1.000000 |
| t42 | 1.000000 |
| t45 | 1.000000 |
| t46 | 1.000000 |
| t47 | 1.000000 |
| t48 | 1.000000 |
| t49 | 1.000000 |
| t50 | 1.000000 |
| w1 | 1.000000 |
| w6 | 1.000000 |
| w7 | 1.000000 |
| w8 | 1.000000 |
| w9 | 1.000000 |
| w10 | 1.000000 |

| | |
|-----|----------|
| w12 | 1.000000 |
| w13 | 1.000000 |
| w14 | 1.000000 |
| w15 | 1.000000 |
| w16 | 1.000000 |
| w17 | 1.000000 |
| w18 | 1.000000 |
| w19 | 1.000000 |
| w24 | 1.000000 |
| w25 | 1.000000 |
| w26 | 1.000000 |
| w27 | 1.000000 |
| w28 | 1.000000 |
| w30 | 1.000000 |
| w32 | 1.000000 |
| w33 | 1.000000 |
| w34 | 1.000000 |
| w36 | 1.000000 |
| w37 | 1.000000 |
| w39 | 1.000000 |
| w40 | 1.000000 |
| w41 | 1.000000 |
| w42 | 1.000000 |
| w43 | 1.000000 |
| w44 | 1.000000 |
| w45 | 1.000000 |
| w46 | 1.000000 |
| w49 | 1.000000 |
| r1 | 1.000000 |
| r2 | 1.000000 |
| r7 | 1.000000 |
| r8 | 1.000000 |
| r9 | 1.000000 |
| r10 | 1.000000 |
| r12 | 1.000000 |
| r13 | 1.000000 |
| r15 | 1.000000 |
| r19 | 1.000000 |
| r21 | 1.000000 |
| r22 | 1.000000 |
| r23 | 1.000000 |
| r26 | 1.000000 |
| r27 | 1.000000 |
| r28 | 1.000000 |
| r29 | 1.000000 |
| r30 | 1.000000 |
| r31 | 1.000000 |
| r33 | 1.000000 |
| r34 | 1.000000 |
| r35 | 1.000000 |
| r36 | 1.000000 |
| r37 | 1.000000 |
| r38 | 1.000000 |
| r39 | 1.000000 |
| r40 | 1.000000 |

| | |
|-----|----------|
| r41 | 1.000000 |
| r42 | 1.000000 |
| r44 | 1.000000 |
| r45 | 1.000000 |
| r46 | 1.000000 |
| r47 | 1.000000 |
| r50 | 1.000000 |
| f1 | 1.000000 |
| f2 | 1.000000 |
| f3 | 1.000000 |
| f4 | 1.000000 |
| f5 | 1.000000 |
| f8 | 1.000000 |
| f9 | 1.000000 |
| f10 | 1.000000 |
| f11 | 1.000000 |
| f12 | 1.000000 |
| f13 | 1.000000 |
| f14 | 1.000000 |
| f15 | 1.000000 |
| f16 | 1.000000 |
| f17 | 1.000000 |
| f18 | 1.000000 |
| f19 | 1.000000 |
| f21 | 1.000000 |
| f22 | 1.000000 |
| f23 | 1.000000 |
| f24 | 1.000000 |
| f28 | 1.000000 |
| f30 | 1.000000 |
| f31 | 1.000000 |
| f34 | 1.000000 |
| f35 | 1.000000 |
| f36 | 1.000000 |
| f37 | 1.000000 |
| f38 | 1.000000 |
| f41 | 1.000000 |
| f44 | 1.000000 |
| f46 | 1.000000 |
| f47 | 1.000000 |
| f48 | 1.000000 |
| f50 | 1.000000 |
| s2 | 1.000000 |
| s3 | 1.000000 |
| s5 | 1.000000 |
| s7 | 1.000000 |
| s8 | 1.000000 |
| s9 | 1.000000 |
| s11 | 1.000000 |
| s14 | 1.000000 |
| s15 | 1.000000 |
| s18 | 1.000000 |
| s20 | 1.000000 |
| s24 | 1.000000 |
| s25 | 1.000000 |

| | |
|-----|----------|
| s27 | 1.000000 |
| s31 | 1.000000 |
| s32 | 1.000000 |
| s34 | 1.000000 |
| s35 | 1.000000 |
| s37 | 1.000000 |
| s40 | 1.000000 |
| s41 | 1.000000 |
| s43 | 1.000000 |
| s49 | 1.000000 |
| s50 | 1.000000 |
| n3 | 1.000000 |
| n7 | 1.000000 |
| n9 | 1.000000 |
| n11 | 1.000000 |
| n15 | 1.000000 |
| n16 | 1.000000 |
| n17 | 1.000000 |
| n26 | 1.000000 |
| n33 | 1.000000 |
| n39 | 1.000000 |
| n42 | 1.000000 |
| n44 | 1.000000 |
| n49 | 1.000000 |

All other variables in the range 1-10100 are zero.

Problem 'Run2nflp.lp' read.

Read time = 0.04 sec.

Tried aggregator 1 time.

LP Presolve eliminated 350 rows and 0 columns.

Reduced LP has 645 rows, 10100 columns, and 20550 nonzeros.

Presolve time = 0.02 sec.

Initializing dual steep norms . . .

Iteration log . . .

| | | | | |
|------------|---|----------------|---|--------------|
| Iteration: | 1 | Dual objective | = | 38609.380000 |
|------------|---|----------------|---|--------------|

Dual simplex - Optimal: Objective = 3.8495660000e+04

Solution time = 0.04 sec. Iterations = 129 (0)

| Variable Name | Solution Value |
|---------------|----------------|
| s1e26 | 1.000000 |
| s2e47 | 1.000000 |
| s3e37 | 1.000000 |
| s4e29 | 1.000000 |
| s5e50 | 1.000000 |
| s6e31 | 1.000000 |
| s7e2 | 1.000000 |
| s8e39 | 1.000000 |
| s9e10 | 1.000000 |
| s10e11 | 1.000000 |
| s11e3 | 1.000000 |
| s12e24 | 1.000000 |
| s13e35 | 1.000000 |

| | |
|--------|----------|
| s14e18 | 1.000000 |
| s15e5 | 1.000000 |
| s16e25 | 1.000000 |
| s17e14 | 1.000000 |
| s18e21 | 1.000000 |
| s19e19 | 1.000000 |
| s20e8 | 1.000000 |
| s21e43 | 1.000000 |
| s22e43 | 1.000000 |
| s23e46 | 1.000000 |
| s24e48 | 1.000000 |
| s25e31 | 1.000000 |
| s26e19 | 1.000000 |
| s27e33 | 1.000000 |
| s28e45 | 1.000000 |
| s29e41 | 1.000000 |
| s30e40 | 1.000000 |
| s31e30 | 1.000000 |
| s32e25 | 1.000000 |
| s33e32 | 1.000000 |
| s34e10 | 1.000000 |
| s35e2 | 1.000000 |
| s36e23 | 1.000000 |
| s37e6 | 1.000000 |
| s38e21 | 1.000000 |
| s39e14 | 1.000000 |
| s40e9 | 1.000000 |
| s41e18 | 1.000000 |
| s42e5 | 1.000000 |
| s43e42 | 1.000000 |
| s44e26 | 1.000000 |
| s45e27 | 1.000000 |
| s46e28 | 1.000000 |
| s47e47 | 1.000000 |
| s48e50 | 1.000000 |
| s49e49 | 1.000000 |
| s50e39 | 1.000000 |
| s51e4 | 1.000000 |
| s52e3 | 1.000000 |
| s53e11 | 1.000000 |
| s54e37 | 1.000000 |
| s55e16 | 1.000000 |
| s56e42 | 1.000000 |
| s57e30 | 1.000000 |
| s58e49 | 1.000000 |
| s59e48 | 1.000000 |
| s60e12 | 1.000000 |
| s61e18 | 1.000000 |
| s62e7 | 1.000000 |
| s63e41 | 1.000000 |
| s64e32 | 1.000000 |
| s65e46 | 1.000000 |
| s66e6 | 1.000000 |
| s67e3 | 1.000000 |
| s68e1 | 1.000000 |

| | |
|---------|----------|
| s69e10 | 1.000000 |
| s70e11 | 1.000000 |
| s71e33 | 1.000000 |
| s72e15 | 1.000000 |
| s73e44 | 1.000000 |
| s74e39 | 1.000000 |
| s75e36 | 1.000000 |
| s76e28 | 1.000000 |
| s77e8 | 1.000000 |
| s78e37 | 1.000000 |
| s79e9 | 1.000000 |
| s80e17 | 1.000000 |
| s81e22 | 1.000000 |
| s82e13 | 1.000000 |
| s83e40 | 1.000000 |
| s84e43 | 1.000000 |
| s85e14 | 1.000000 |
| s86e26 | 1.000000 |
| s87e25 | 1.000000 |
| s88e34 | 1.000000 |
| s89e35 | 1.000000 |
| s90e13 | 1.000000 |
| s91e39 | 1.000000 |
| s92e41 | 1.000000 |
| s93e19 | 1.000000 |
| s94e10 | 1.000000 |
| s95e40 | 1.000000 |
| s96e42 | 1.000000 |
| s97e45 | 1.000000 |
| s98e30 | 1.000000 |
| s99e36 | 1.000000 |
| s100e33 | 1.000000 |
| s101e43 | 1.000000 |
| s102e44 | 1.000000 |
| s103e4 | 1.000000 |
| s104e2 | 1.000000 |
| s105e17 | 1.000000 |
| s106e7 | 1.000000 |
| s107e6 | 1.000000 |
| s108e12 | 1.000000 |
| s109e35 | 1.000000 |
| s110e23 | 1.000000 |
| s111e29 | 1.000000 |
| s112e31 | 1.000000 |
| s113e46 | 1.000000 |
| s114e38 | 1.000000 |
| s115e28 | 1.000000 |
| s116e50 | 1.000000 |
| s117e26 | 1.000000 |
| s118e34 | 1.000000 |
| s119e15 | 1.000000 |
| s120e8 | 1.000000 |
| s121e20 | 1.000000 |
| s122e37 | 1.000000 |
| s123e22 | 1.000000 |

| | |
|---------|----------|
| s124e34 | 1.000000 |
| s125e44 | 1.000000 |
| s126e28 | 1.000000 |
| s127e31 | 1.000000 |
| s128e38 | 1.000000 |
| s129e35 | 1.000000 |
| s130e50 | 1.000000 |
| s131e30 | 1.000000 |
| s132e11 | 1.000000 |
| s133e9 | 1.000000 |
| s134e19 | 1.000000 |
| s135e20 | 1.000000 |
| s136e37 | 1.000000 |
| s137e8 | 1.000000 |
| s138e49 | 1.000000 |
| s139e41 | 1.000000 |
| s140e3 | 1.000000 |
| s141e1 | 1.000000 |
| s142e23 | 1.000000 |
| s143e48 | 1.000000 |
| s144e22 | 1.000000 |
| s145e21 | 1.000000 |
| s146e10 | 1.000000 |
| s147e14 | 1.000000 |
| s148e2 | 1.000000 |
| s149e18 | 1.000000 |
| s150e15 | 1.000000 |
| s151e36 | 1.000000 |
| s152e12 | 1.000000 |
| s153e24 | 1.000000 |
| s154e46 | 1.000000 |
| s155e13 | 1.000000 |
| s156e6 | 1.000000 |
| s157e47 | 1.000000 |
| s158e5 | 1.000000 |
| s159e15 | 1.000000 |
| s160e40 | 1.000000 |
| s161e5 | 1.000000 |
| s162e2 | 1.000000 |
| s163e34 | 1.000000 |
| s164e35 | 1.000000 |
| s165e9 | 1.000000 |
| s166e50 | 1.000000 |
| s167e11 | 1.000000 |
| s168e41 | 1.000000 |
| s169e32 | 1.000000 |
| s170e19 | 1.000000 |
| s171e43 | 1.000000 |
| s172e25 | 1.000000 |
| s173e7 | 1.000000 |
| s174e49 | 1.000000 |
| s175e30 | 1.000000 |
| s176e8 | 1.000000 |
| s177e31 | 1.000000 |
| s178e14 | 1.000000 |

| | |
|---------|----------|
| s179e33 | 1.000000 |
| s180e20 | 1.000000 |
| s181e27 | 1.000000 |
| s182e24 | 1.000000 |
| s183e33 | 1.000000 |
| s184e3 | 1.000000 |
| s185e9 | 1.000000 |
| s186e34 | 1.000000 |
| s187e15 | 1.000000 |
| s188e16 | 1.000000 |
| s189e17 | 1.000000 |
| s190e44 | 1.000000 |
| s191e39 | 1.000000 |
| s192e7 | 1.000000 |
| s193e26 | 1.000000 |
| s194e42 | 1.000000 |
| s195e49 | 1.000000 |
| m2 | 1.000000 |
| m3 | 1.000000 |
| m5 | 1.000000 |
| m8 | 1.000000 |
| m10 | 1.000000 |
| m11 | 1.000000 |
| m14 | 1.000000 |
| m18 | 1.000000 |
| m19 | 1.000000 |
| m21 | 1.000000 |
| m24 | 1.000000 |
| m25 | 1.000000 |
| m26 | 1.000000 |
| m29 | 1.000000 |
| m31 | 1.000000 |
| m35 | 1.000000 |
| m37 | 1.000000 |
| m39 | 1.000000 |
| m43 | 1.000000 |
| m47 | 1.000000 |
| m50 | 1.000000 |
| t2 | 1.000000 |
| t3 | 1.000000 |
| t4 | 1.000000 |
| t5 | 1.000000 |
| t6 | 1.000000 |
| t9 | 1.000000 |
| t10 | 1.000000 |
| t11 | 1.000000 |
| t14 | 1.000000 |
| t16 | 1.000000 |
| t18 | 1.000000 |
| t19 | 1.000000 |
| t21 | 1.000000 |
| t23 | 1.000000 |
| t25 | 1.000000 |
| t26 | 1.000000 |
| t27 | 1.000000 |

| | |
|-----|----------|
| t28 | 1.000000 |
| t30 | 1.000000 |
| t31 | 1.000000 |
| t32 | 1.000000 |
| t33 | 1.000000 |
| t37 | 1.000000 |
| t39 | 1.000000 |
| t40 | 1.000000 |
| t41 | 1.000000 |
| t42 | 1.000000 |
| t43 | 1.000000 |
| t45 | 1.000000 |
| t46 | 1.000000 |
| t47 | 1.000000 |
| t48 | 1.000000 |
| t49 | 1.000000 |
| t50 | 1.000000 |
| w1 | 1.000000 |
| w3 | 1.000000 |
| w6 | 1.000000 |
| w7 | 1.000000 |
| w8 | 1.000000 |
| w9 | 1.000000 |
| w10 | 1.000000 |
| w11 | 1.000000 |
| w12 | 1.000000 |
| w13 | 1.000000 |
| w14 | 1.000000 |
| w15 | 1.000000 |
| w17 | 1.000000 |
| w18 | 1.000000 |
| w22 | 1.000000 |
| w25 | 1.000000 |
| w26 | 1.000000 |
| w28 | 1.000000 |
| w30 | 1.000000 |
| w32 | 1.000000 |
| w33 | 1.000000 |
| w34 | 1.000000 |
| w35 | 1.000000 |
| w36 | 1.000000 |
| w37 | 1.000000 |
| w39 | 1.000000 |
| w40 | 1.000000 |
| w41 | 1.000000 |
| w42 | 1.000000 |
| w43 | 1.000000 |
| w44 | 1.000000 |
| w46 | 1.000000 |
| w48 | 1.000000 |
| w49 | 1.000000 |
| r2 | 1.000000 |
| r4 | 1.000000 |
| r6 | 1.000000 |
| r7 | 1.000000 |

| | |
|-----|----------|
| r8 | 1.000000 |
| r10 | 1.000000 |
| r12 | 1.000000 |
| r13 | 1.000000 |
| r15 | 1.000000 |
| r17 | 1.000000 |
| r19 | 1.000000 |
| r20 | 1.000000 |
| r22 | 1.000000 |
| r23 | 1.000000 |
| r26 | 1.000000 |
| r28 | 1.000000 |
| r29 | 1.000000 |
| r30 | 1.000000 |
| r31 | 1.000000 |
| r33 | 1.000000 |
| r34 | 1.000000 |
| r35 | 1.000000 |
| r36 | 1.000000 |
| r37 | 1.000000 |
| r38 | 1.000000 |
| r39 | 1.000000 |
| r40 | 1.000000 |
| r41 | 1.000000 |
| r42 | 1.000000 |
| r43 | 1.000000 |
| r44 | 1.000000 |
| r45 | 1.000000 |
| r46 | 1.000000 |
| r50 | 1.000000 |
| f1 | 1.000000 |
| f2 | 1.000000 |
| f3 | 1.000000 |
| f5 | 1.000000 |
| f6 | 1.000000 |
| f8 | 1.000000 |
| f9 | 1.000000 |
| f10 | 1.000000 |
| f11 | 1.000000 |
| f12 | 1.000000 |
| f13 | 1.000000 |
| f14 | 1.000000 |
| f15 | 1.000000 |
| f18 | 1.000000 |
| f19 | 1.000000 |
| f20 | 1.000000 |
| f21 | 1.000000 |
| f22 | 1.000000 |
| f23 | 1.000000 |
| f24 | 1.000000 |
| f28 | 1.000000 |
| f30 | 1.000000 |
| f31 | 1.000000 |
| f34 | 1.000000 |
| f35 | 1.000000 |

| | |
|-----|----------|
| f36 | 1.000000 |
| f37 | 1.000000 |
| f38 | 1.000000 |
| f41 | 1.000000 |
| f44 | 1.000000 |
| f46 | 1.000000 |
| f47 | 1.000000 |
| f48 | 1.000000 |
| f49 | 1.000000 |
| f50 | 1.000000 |
| s2 | 1.000000 |
| s5 | 1.000000 |
| s7 | 1.000000 |
| s8 | 1.000000 |
| s9 | 1.000000 |
| s11 | 1.000000 |
| s14 | 1.000000 |
| s15 | 1.000000 |
| s19 | 1.000000 |
| s20 | 1.000000 |
| s24 | 1.000000 |
| s25 | 1.000000 |
| s27 | 1.000000 |
| s30 | 1.000000 |
| s31 | 1.000000 |
| s32 | 1.000000 |
| s33 | 1.000000 |
| s34 | 1.000000 |
| s35 | 1.000000 |
| s40 | 1.000000 |
| s41 | 1.000000 |
| s43 | 1.000000 |
| s49 | 1.000000 |
| s50 | 1.000000 |
| n3 | 1.000000 |
| n7 | 1.000000 |
| n9 | 1.000000 |
| n15 | 1.000000 |
| n16 | 1.000000 |
| n17 | 1.000000 |
| n26 | 1.000000 |
| n33 | 1.000000 |
| n34 | 1.000000 |
| n39 | 1.000000 |
| n42 | 1.000000 |
| n44 | 1.000000 |
| n49 | 1.000000 |

All other variables in the range 1-10100 are zero.

APPENDIX C

CHAPTER FOUR VISUAL BASIC CODE

```
Private Sub CommandButton1_Click()
```

```
'This subroutine finds and writes the preference for the employees  
'on the Employees page
```

```
'Shane Knighton 06-09-05
```

```
'Revised 07-13-05 to contain weekend shift restrictions
```

```
'Dimension counting variables for looping  
Dim i, j, k As Integer
```

```
'PROBLEM DATA: RETRIEVED FROM THE WORKSHEETS
```

```
'SHIFTS WORKSHEET
```

```
'Get number of shifts, days, and weeks to schedule  
Dim NumShifts As Integer  
NumShifts = Sheets("Shifts").Cells(2, 6).Value
```

```
'EMPLOYEE WORKSHEET
```

```
'Get number of employees  
Dim NumEmp As Integer  
NumEmp = Sheets("Employees").Cells(4, 2)  
'Get the employee ranking deviation and the deviation within each  
'employee's shifts  
Dim EmpDevPer, EmpShiftDevPer As Single  
EmpDevPer = Sheets("Employees").Cells(4, 19)  
EmpShiftDevPer = Sheets("Employees").Cells(5, 23)
```

```
'Calculate the rankings  
Dim HighRank, EmpDev, EmpShiftDev As Single  
HighRank = NumEmp * 100  
EmpDev = HighRank * EmpDevPer  
EmpShiftDev = EmpDev * EmpShiftDevPer
```

```
'Fill in the employee preferences on the "Employee" sheet  
Dim rank As Single  
Dim putrow As Integer  
Range(Cells(10, 4), Cells(9 + NumShifts, 4 + NumEmp)).Clear  
For j = 1 To NumEmp  
    For i = 1 To NumShifts  
        rank = HighRank - EmpDev * (j - 1)  
        putrow = Int((NumShifts * Rnd()))  
        If putrow < 1 Then putrow = 126
```



```

        Sheets("Employees").Cells(9 + putrow, 3 + j).Value = rank + EmpShiftDev *
    (i - 1)
    Next i
Next j

End Sub

```

```

Private Sub CommandButton2_Click()
'This subroutine will write out in CPLEX .LP format
'a LP for solving large crew/nurse scheduling problems
'using a network-based linear program. The network is designed
'to do tour scheduling problems.
'Shane Knighton 6-09-05
'Rev 06-10-5

```

```

'Dimension counting variables for looping
Dim i, j, k As Integer

```

```

'PROBLEM DATA: RETRIEVED FROM THE WORKSHEETS

```

```

'SHIFTS WORKSHEET
'Get number of shifts, days, and weeks to schedule
Dim NumShifts, NumShiftsPerDay, NumDaysPerWeek, NumWeeks As Integer
NumShiftsPerDay = Sheets("Shifts").Cells(3, 2).Value
NumDaysPerWeek = Sheets("Shifts").Cells(5, 2).Value
NumWeeks = Sheets("Shifts").Cells(2, 4).Value
NumShifts = Sheets("Shifts").Cells(2, 6).Value
'Min and Max shifts per week per employee
Dim MaxNumShifts, MinNumShifts As Integer
MaxNumShifts = Sheets("Shifts").Cells(4, 4).Value
MinNumShifts = Sheets("Shifts").Cells(5, 4).Value
'ShiftPref is a NumShift by 1 matrix that stores
'the preference for each shift to be selected
'to cover the hourly demand from the HourDemMatrix
Dim ShiftPref() As Single
ReDim ShiftPref(1 To NumShifts)
For i = 1 To NumShifts
    ShiftPref(i) = Sheets("Shifts").Cells(9 + i, 7).Value
Next i
'Find the shift starting hour and length
Dim ShiftStartHour() As Integer
ReDim ShiftStartHour(1 To NumShifts, 1 To NumShifts) As Integer
For i = 1 To NumShifts

```

```

    ShiftStartHour(i, 1) = Sheets("Shifts").Cells(9 + i, 3).Value
    ShiftStartHour(i, 2) = Sheets("Shifts").Cells(9 + i, 5).Value
Next i
'Get the length of the lunch break
Dim LunchLen As Integer
LunchLen = Sheets("Shifts").Cells(5, 6).Value
'Get the demand for each shift
Dim ShiftDemand() As Integer
ReDim ShiftDemand(1 To NumShifts) As Integer
For i = 1 To NumShifts
    ShiftDemand(i) = Sheets("Shifts").Cells(9 + i, 8).Value
Next i

'CONFLICTS WORKSHEET
'Get the maximum number of conflicted shifts
Dim MaxNumConfl As Integer
MaxNumConfl = Sheets("Conflicts").Cells(5, 2).Value
'ConflictMatrix is a NumShift by maximum number of conflicts
'matrix that has the shift id of a conflicted shift in row of
'each shift that has any conflicted shifts. The row of the matrix
'is empty if the shift id corresponding to that shift does not
'have any conflicted shifts.
Dim ConflictMatrix() As Integer
ReDim ConflictMatrix(1 To NumShifts, 1 To MaxNumConfl) As Integer
'Populate ConflictMatrix
For i = 1 To NumShifts
    If Sheets("Conflicts").Cells(9 + i, 2) <> 0 Then
        For j = 1 To MaxNumConfl
            ConflictMatrix(i, j) = Sheets("Conflicts").Cells(9 + i, 1 + j)
        Next j
    End If
Next i

'EMPLOYEE WORKSHEET
'Get number of employees
Dim NumEmp As Integer
NumEmp = Sheets("Employees").Cells(4, 2)
'PriorityMatrix is a NumShifts by NumEmployee matrix
'containing the priority of each shift for each employee
Dim PriorityMatrix() As Single
ReDim PriorityMatrix(1 To NumShifts, 1 To NumEmp) As Single
'Populate PriorityMatrix
For i = 1 To NumShifts

```

```

    For j = 1 To NumEmp
        PriorityMatrix(i, j) = Sheets("Employees").Cells(9 + i, 3 + j)
    Next j
Next i

```

```

'OPEN FILE TO WRITE .LP TO
'Get Name of Problem
Dim Name, filepath As String
Name = "w" & NumWeeks & "e" & NumEmp
'Open a file to write to
filepath = "C:\Documents and Settings\Shane\My Documents\Dissertation\Phase
4\" & Name & ".lp"
Open filepath For Output As #1

```

```

*****

```

```

'BEGIN WRITE LP TO FILE

```

```

'WRITE HEADER AND OBJECTIVE FUNCTION
'VarName is the concatonized shift and employee variable
Print #1, "MAXIMIZE"
Print #1, "Z:"

```

```

'Write employee preferences into objective function
Dim varName As String
Dim EmpAvailable As Boolean
Dim dum1 As Integer
'i is the loop for shifts
For i = 1 To NumShifts
    'j is the loop for employees
    For j = 1 To NumEmp
        'Check if variable should be included
        'If preference is zero then employee is unavailable
        'and variable should not be in model
        If PriorityMatrix(i, j) = 0 Then EmpAvailable = False
        'Skip writing the variable to the file if EmpAvailable is false
        If EmpAvailable = True Then
            'Concatenate variable name
            varName = "s" & i & "e" & j
            'Write variables and cost coefficients to file
            'Write final variable and cost to file, leave off the "+"
            If i = NumShifts And j = NumEmp Then
                Print #1, PriorityMatrix(i, j) & " " & varName
            End If
        End If
    Next j
Next i

```

```

Else
    Print #1, PriorityMatrix(i, j) & " " & varName & " + "
End If
End If
'If the final employee is unavailable need to write the ending to file
'use a dummy variable to keep formatting
If EmpAvailable = False And i = NumShifts And j = NumEmp Then
    Print #1, " dum1"
End If
'Set availability to true
EmpAvailable = True
Next j
Next i

```

'WRITE CONSTRAINTS

'Header

Print #1, "SUBJECT TO"

'Constraint number counter, needs to be double because
'of the large number of constraints

Dim ConstNum As Double

ConstNum = 1

'Define the 9 sets of constraints

' i = loop for shifts

' j = loop for employees

' k = loop for weeks

' l = loop for days

'Set1: shifts = demand for that shift determined in Set 0

' $s(i)e(1) + \dots + s(i)e(\text{NumEmp}) = \text{ShiftDemand}(i)$

'Set2: total shifts per employee for each day

' of each week

' will be required to be less than one

' by Set 5, needed to total number

' of shifts per week

' $s(i)e(j) + \dots + s(\text{NumShiftsPerDay})e(j) = e(j)w(k)d(l)$

'Set3: each employee has less than max number of shifts per week

' $e(j)w(k)d(l) + \dots + e(j)w(k)d(\text{NumDaysPerWeek}) \leq \text{MaxNumShifts}$

'Set4: each employee has more than min number of shifts per week

' $e(j)w(k)d(l) + \dots + e(j)w(k)d(\text{NumDaysPerWeek}) \geq \text{MinNumShifts}$

'Set5: only one shift per day per employee

' $e(j)w(k)d(l) \leq 1$

'Set6: shift confliction for each employee

' example if shifts 25 and 26 are conflicted

' $s025e(j) + s026e(j) \leq 1$

```

'Set7: each employee has less than max number of weekend shifts
'   e(j)w(k)d(6)+ e(j)w(k)d(7) + ... + e(j)w(k+4)d(6) + e(j)w(k)d(7)
<=MaxWeekendShifts

'Write Set1 constraints
Dim dum2 As Integer
'i is the loop for shifts
For i = 1 To NumShifts
    'Write constraint lable
    Print #1, "c" & ConstNum & ":"
    'j is the loop for employees
    For j = 1 To NumEmp
        'Check if variable should be included
        'If preference is zero then employee is unavailable
        'and variable should not be in model
        If PriorityMatrix(i, j) = 0 Then EmpAvailable = False
        'Skip writing the variable to the file if EmpAvailable is false
        If EmpAvailable = True Then
            'Concatenate variable name
            varName = "s" & i & "e" & j
            'Write variables and coefficients to file, leave off "+" for final emp
            If j = NumEmp Then
                Print #1, varName & " = " & ShiftDemand(i)
            Else
                Print #1, varName & " + "
            End If
        End If
        'If the final employee is unavailable need to write the = ShiftDemand to file
        'use a dummy variable to keep formatting
        If EmpAvailable = False And j = NumEmp Then
            Print #1, " dum2 = " & ShiftDemand(i)
        End If
        'Set availability to true
        EmpAvailable = True
    Next j
    ConstNum = ConstNum + 1
Next i

'Write Set2 constraints
Dim DayWeekVarName As String
Dim dum3 As Integer
'Initialize i, day and week
i = 1
k = 1

```

```

l = 1
'i is the loop for shifts
Do While i < NumShifts
    'j is the loop for employees
    For j = 1 To NumEmp
        'Write constraint lable
        Print #1, "c" & ConstNum & ":"
        'ii is the nested loop for shifts
        For ii = i To i + (NumShiftsPerDay - 1)
            'Check if variable should be included
            'If preference is zero then employee is unavailable
            'and variable should not be in model
            If PriorityMatrix(ii, j) = 0 Then EmpAvailable = False
            'Skip writing the variable to the file if EmpAvailable is false
            If EmpAvailable = True Then
                'Concatenate variable names
                varName = "s" & ii & "e" & j
                DayWeekVarName = "em" & j & "w" & k & "d" & l
                'Write variables and coefficients to file
                If ii = i + (NumShiftsPerDay - 1) Then
                    Print #1, varName & " - " & DayWeekVarName & " = 0"
                Else
                    Print #1, varName & " + "
                End If
            End If
            'If the final employee is unavailable need to write the = ShiftDemand to
file
            'use a dummy variable to keep formatting
            If EmpAvailable = False And ii = i + (NumShiftsPerDay - 1) Then
                Print #1, " dum3 - " & DayWeekVarName & " = 0"
            End If
            'Set availability to true
            EmpAvailable = True
        Next ii
        ConstNum = ConstNum + 1
    Next j
    i = i + NumShiftsPerDay
    If l < NumDaysPerWeek Then
        l = l + 1
    ElseIf l = NumShiftsPerDay Then
        k = k + 1
        l = 1
    End If
    Loop 'End loop for Set2 Constraints

```

```

'Write Set3 constraints
'j is loop for employees
For j = 1 To NumEmp
  Print #1, "c" & ConstNum & ":"
  'k is loop for weeks
  For k = 1 To NumWeeks
    'l is loop for days
    For l = 1 To NumDaysPerWeek
      DayWeekVarName = "em" & j & "w" & k & "d" & l
      If l = NumDaysPerWeek Then
        Print #1, DayWeekVarName & " <= " & MaxNumShifts
      Else
        Print #1, DayWeekVarName & " + "
      End If
    Next l
  Next k
  ConstNum = ConstNum + 1
Next j

```

```

'Write Set4 constraints
'j is loop for employees
For j = 1 To NumEmp
  Print #1, "c" & ConstNum & ":"
  'k is loop for weeks
  For k = 1 To NumWeeks
    'l is loop for days
    For l = 1 To NumDaysPerWeek
      DayWeekVarName = "em" & j & "w" & k & "d" & l
      If l = NumDaysPerWeek Then
        Print #1, DayWeekVarName & " >= " & MinNumShifts
      Else
        Print #1, DayWeekVarName & " + "
      End If
    Next l
  Next k
  ConstNum = ConstNum + 1
Next j

```

```

'Write Set5 constraints
'j is loop for employees
For j = 1 To NumEmp
  Print #1, "c" & ConstNum & ":"
  'k is loop for weeks

```

```

For k = 1 To NumWeeks
    'l is loop for days
    For l = 1 To NumDaysPerWeek
        DayWeekVarName = "em" & j & "w" & k & "d" & l
        Print #1, DayWeekVarName & " <= 1"
    Next l
Next k
ConstNum = ConstNum + 1
Next j

'Write Set6 constraints
For i = 1 To NumShifts
    For k = 1 To MaxNumConfl
        If ConflictMatrix(i, k) <> 0 Then
            For j = 1 To NumEmp
                'Check if variable should be included
                'If preference is zero for either shift then employee is unavailable
                'and variable should not be in model
                If PriorityMatrix(i, j) = 0 Then EmpAvailable = False
                If PriorityMatrix(i, k) = 0 Then EmpAvailable = False
                'Skip writing the constraint to the file if EmpAvailable is false
                If EmpAvailable = True Then
                    Print #1, "c" & ConstNum & ":"
                    Print #1, "s" & i & "e" & j & " + s" & ConflictMatrix(i, k) & "e" & j & "
<= 1"
                    ConstNum = ConstNum + 1
                End If
                'Set availability to true
                EmpAvailable = True
            Next j
        End If
    Next k
Next i

'Write Set7 constraints
'Get maximum number of weekend shifts per month
Dim NumWeekendShifts, NumMonths, m As Integer
NumWeekendShifts = Sheets("Shifts").Cells(4, 8).Value
'j is loop for employees
For j = 1 To NumEmp
    Print #1, "c" & ConstNum & ":"
    'k is loop for weeks
    'm is loop for months
    NumMonths = Int(NumWeeks / 4)

```



```

For m = 1 To NumMonths
  For k = 1 To 4
    'l is loop for days
    For l = 6 To 7
      If l = 6 Then DayWeekVarName = "em" & j & "w" & m * k & "d" & (m * k *
7) - 1
      If l = 7 Then DayWeekVarName = "em" & j & "w" & m * k & "d" & (m * k *
7)
      If k * l = 4 * 7 Then
        Print #1, DayWeekVarName & " <= " & NumWeekendShifts
      Else
        Print #1, DayWeekVarName & " + "
      End If
    Next l
  Next k
Next m
ConstNum = ConstNum + 1
Next j

```

```

'Set all dummy variables to 0
Print #1, "dum1 = 0"
Print #1, "dum2 = 0"
Print #1, "dum3 = 0"

```

```

'Write END
Print #1, "END"
'Close mps file
Close #1

```

```
Exit Sub
```

```

ErrorHandler:
MsgBox ("Error")
'Close mps file
Close #1

```

```
End Sub
```

```
Private Sub CommandButton3_Click()
```

```

'This subroutine will write out in CPLEX .LP format
'a LP for solving large crew/nurse scheduling problems

```

'using a network-based linear program. The network is designed
'to do tour scheduling problems.

'Shane Knighton 6-09-05

'Rev 06-10-5 - Writes integer constraints

'Dimension counting variables for looping
Dim i, j, k As Integer

'PROBLEM DATA: RETRIEVED FROM THE WORKSHEETS

'SHIFTS WORKSHEET

'Get number of shifts, days, and weeks to schedule

Dim NumShifts, NumShiftsPerDay, NumDaysPerWeek, NumWeeks As Integer

NumShiftsPerDay = Sheets("Shifts").Cells(3, 2).Value

NumDaysPerWeek = Sheets("Shifts").Cells(5, 2).Value

NumWeeks = Sheets("Shifts").Cells(2, 4).Value

NumShifts = Sheets("Shifts").Cells(2, 6).Value

'Min and Max shifts per week per employee

Dim MaxNumShifts, MinNumShifts As Integer

MaxNumShifts = Sheets("Shifts").Cells(4, 4).Value

MinNumShifts = Sheets("Shifts").Cells(5, 4).Value

'ShiftPref is a NumShift by 1 matrix that stores

'the preference for each shift to be selected

'to cover the hourly demand from the HourDemMatrix

Dim ShiftPref() As Single

ReDim ShiftPref(1 To NumShifts)

For i = 1 To NumShifts

 ShiftPref(i) = Sheets("Shifts").Cells(9 + i, 7).Value

Next i

'Find the shift starting hour and length

Dim ShiftStartHour() As Integer

ReDim ShiftStartHour(1 To NumShifts, 1 To NumShifts) As Integer

For i = 1 To NumShifts

 ShiftStartHour(i, 1) = Sheets("Shifts").Cells(9 + i, 3).Value

 ShiftStartHour(i, 2) = Sheets("Shifts").Cells(9 + i, 5).Value

Next i

'Get the length of the lunch break

Dim LunchLen As Integer

LunchLen = Sheets("Shifts").Cells(5, 6).Value

'Get the demand for each shift

Dim ShiftDemand() As Integer

ReDim ShiftDemand(1 To NumShifts) As Integer

For i = 1 To NumShifts

```

    ShiftDemand(i) = Sheets("Shifts").Cells(9 + i, 8).Value
Next i

```

'CONFLICTS WORKSHEET

```

'Get the maximum number of conflicted shifts
Dim MaxNumConfl As Integer
MaxNumConfl = Sheets("Conflicts").Cells(5, 2).Value
'ConflictMatrix is a NumShift by maximum number of conflicts
'matrix that has the shift id of a conflicted shift in row of
'each shift that has any conflicted shifts. The row of the matrix
'is empty if the shift id corresponding to that shift does not
'have any conflicted shifts.
Dim ConflictMatrix() As Integer
ReDim ConflictMatrix(1 To NumShifts, 1 To MaxNumConfl) As Integer
'Populate ConflictMatrix
For i = 1 To NumShifts
    If Sheets("Conflicts").Cells(9 + i, 2) <> 0 Then
        For j = 1 To MaxNumConfl
            ConflictMatrix(i, j) = Sheets("Conflicts").Cells(9 + i, 1 + j)
        Next j
    End If
Next i

```

'EMPLOYEE WORKSHEET

```

'Get number of employees
Dim NumEmp As Integer
NumEmp = Sheets("Employees").Cells(4, 2)
'PriorityMatrix is a NumShifts by NumEmployee matrix
'containing the priority of each shift for each employee
Dim PriorityMatrix() As Single
ReDim PriorityMatrix(1 To NumShifts, 1 To NumEmp) As Single
'Populate PriorityMatrix
For i = 1 To NumShifts
    For j = 1 To NumEmp
        PriorityMatrix(i, j) = Sheets("Employees").Cells(9 + i, 3 + j)
    Next j
Next i

```

'OPEN FILE TO WRITE .LP TO

```

'Get Name of Problem
Dim Name, filepath As String
Name = "w" & NumWeeks & "e" & NumEmp & "ip"
'Open a file to write to

```

```
filepath = "C:\Documents and Settings\Shane\My Documents\Dissertation\Phase
1\New Coding\" & Name & ".lp"
Open filepath For Output As #1
```

```
*****
```

```
'BEGIN WRITE LP TO FILE
```

```
'WRITE HEADER AND OBJECTIVE FUNCTION
```

```
'VarName is the concatenated shift and employee variable
```

```
Print #1, "MAXIMIZE"
```

```
Print #1, "Z:"
```

```
'Write employee preferences into objective function
```

```
Dim varName As String
```

```
Dim EmpAvailable As Boolean
```

```
Dim dum1 As Integer
```

```
'i is the loop for shifts
```

```
For i = 1 To NumShifts
```

```
    'j is the loop for employees
```

```
    For j = 1 To NumEmp
```

```
        'Check if variable should be included
```

```
        'If preference is zero then employee is unavailable
```

```
        'and variable should not be in model
```

```
        If PriorityMatrix(i, j) = 0 Then EmpAvailable = False
```

```
        'Skip writing the variable to the file if EmpAvailable is false
```

```
        If EmpAvailable = True Then
```

```
            'Concatenate variable name
```

```
            varName = "s" & i & "e" & j
```

```
            'Write variables and cost coefficients to file
```

```
            'Write final variable and cost to file, leave off the "+"
```

```
            If i = NumShifts And j = NumEmp Then
```

```
                Print #1, PriorityMatrix(i, j) & " " & varName
```

```
            Else
```

```
                Print #1, PriorityMatrix(i, j) & " " & varName & " + "
```

```
            End If
```

```
        End If
```

```
        'If the final employee is unavailable need to write the ending to file
```

```
        'use a dummy variable to keep formatting
```

```
        If EmpAvailable = False And i = NumShifts And j = NumEmp Then
```

```
            Print #1, " dum1"
```

```
        End If
```

```
        'Set availability to true
```

```

        EmpAvailable = True
    Next j
Next i

'WRITE CONSTRAINTS
'Header
Print #1, "SUBJECT TO"
'Constraint number counter, needs to be double because
'of the large number of constraints
Dim ConstNum As Double
ConstNum = 1

'Define the 9 sets of constraints
' i = loop for shifts
' j = loop for employees
' k = loop for weeks
' l = loop for days
'Set1: shifts = demand for that shift determined in Set 0
'     $s(i)e(1) + \dots + s(i)e(\text{NumEmp}) = \text{ShiftDemand}(i)$ 
'Set2: total shifts per employee for each day
'    of each week
'    will be required to be less than one
'    by Set 5, needed to total number
'    of shifts per week
'     $s(i)e(j) + \dots + s(\text{NumShiftsPerDay})e(j) = e(j)w(k)d(l)$ 
'Set3: each employee has less than max number of shifts per week
'     $e(j)w(k)d(l) + \dots + e(j)w(k)d(\text{NumDaysPerWeek}) \leq \text{MaxNumShifts}$ 
'Set4: each employee has more than min number of shifts per week
'     $e(j)w(k)d(l) + \dots + e(j)w(k)d(\text{NumDaysPerWeek}) \geq \text{MinNumShifts}$ 
'Set5: only one shift per day per employee
'     $e(j)w(k)d(l) \leq 1$ 
'Set6: shift confliction for each employee
'    example if shifts 25 and 26 are conflicted
'     $s_{025}e(j) + s_{026}e(j) \leq 1$ 
'Set7: each employee has less than max number of weekend shifts
'     $e(j)w(k)d(6) + e(j)w(k)d(7) + \dots + e(j)w(k+4)d(6) + e(j)w(k)d(7)$ 
'     $\leq \text{MaxWeekendShifts}$ 

'Write Set1 constraints
Dim dum2 As Integer
'i is the loop for shifts
For i = 1 To NumShifts
    'Write constraint lable
    Print #1, "c" & ConstNum & ":"

```

```

'j is the loop for employees
For j = 1 To NumEmp
    'Check if variable should be included
    'If preference is zero then employee is unavailable
    'and variable should not be in model
    If PriorityMatrix(i, j) = 0 Then EmpAvailable = False
    'Skip writing the variable to the file if EmpAvailable is false
    If EmpAvailable = True Then
        'Concatenate variable name
        varName = "s" & i & "e" & j
        'Write variables and coefficients to file, leave off "+" for final emp
        If j = NumEmp Then
            Print #1, varName & " = " & ShiftDemand(i)
        Else
            Print #1, varName & " + "
        End If
    End If
    'If the final employee is unavailable need to write the = ShiftDemand to file
    'use a dummy variable to keep formatting
    If EmpAvailable = False And j = NumEmp Then
        Print #1, " dum2 = " & ShiftDemand(i)
    End If
    'Set availability to true
    EmpAvailable = True
Next j
ConstNum = ConstNum + 1
Next i

'Write Set2 constraints
Dim DayWeekVarName As String
Dim dum3 As Integer
'Initialize i, day and week
i = 1
k = 1
l = 1
'i is the loop for shifts
Do While i < NumShifts
    'j is the loop for employees
    For j = 1 To NumEmp
        'Write constraint lable
        Print #1, "c" & ConstNum & " :."
        'ii is the nested loop for shifts
        For ii = i To i + (NumShiftsPerDay - 1)
            'Check if variable should be included

```

```

'If preference is zero then employee is unavailable
'and variable should not be in model
If PriorityMatrix(ii, j) = 0 Then EmpAvailable = False
'Skip writing the variable to the file if EmpAvailable is false
If EmpAvailable = True Then
    'Concatenate variable names
    varName = "s" & ii & "e" & j
    DayWeekVarName = "em" & j & "w" & k & "d" & l
    'Write variables and coefficients to file
    If ii = i + (NumShiftsPerDay - 1) Then
        Print #1, varName & " - " & DayWeekVarName & " = 0"
    Else
        Print #1, varName & " + "
    End If
End If
'If the final employee is unavailable need to write the = ShiftDemand to
file
'use a dummy variable to keep formatting
If EmpAvailable = False And ii = i + (NumShiftsPerDay - 1) Then
    Print #1, " dum3 - " & DayWeekVarName & " = 0"
End If
'Set availability to true
EmpAvailable = True
Next ii
ConstNum = ConstNum + 1
Next j
i = i + NumShiftsPerDay
If l < NumDaysPerWeek Then
    l = l + 1
Elseif l = NumShiftsPerDay Then
    k = k + 1
    l = 1
End If
Loop 'End loop for Set2 Constraints

'Write Set3 constraints
'j is loop for employees
For j = 1 To NumEmp
    Print #1, "c" & ConstNum & ":"
    'k is loop for weeks
    For k = 1 To NumWeeks
        'l is loop for days
        For l = 1 To NumDaysPerWeek
            DayWeekVarName = "em" & j & "w" & k & "d" & l

```

```

        If l = NumDaysPerWeek Then
            Print #1, DayWeekVarName & " <= " & MaxNumShifts
        Else
            Print #1, DayWeekVarName & " + "
        End If
    Next l
Next k
ConstNum = ConstNum + 1
Next j

```

```

'Write Set4 constraints
'j is loop for employees
For j = 1 To NumEmp
    Print #1, "c" & ConstNum & ":"
    'k is loop for weeks
    For k = 1 To NumWeeks
        'l is loop for days
        For l = 1 To NumDaysPerWeek
            DayWeekVarName = "em" & j & "w" & k & "d" & l
            If l = NumDaysPerWeek Then
                Print #1, DayWeekVarName & " >= " & MinNumShifts
            Else
                Print #1, DayWeekVarName & " + "
            End If
        Next l
    Next k
    ConstNum = ConstNum + 1
Next j

```

```

'Write Set5 constraints
'j is loop for employees
For j = 1 To NumEmp
    Print #1, "c" & ConstNum & ":"
    'k is loop for weeks
    For k = 1 To NumWeeks
        'l is loop for days
        For l = 1 To NumDaysPerWeek
            DayWeekVarName = "em" & j & "w" & k & "d" & l
            Print #1, DayWeekVarName & " <= 1"
        Next l
    Next k
    ConstNum = ConstNum + 1
Next j

```



```

'Write Set6 constraints
For i = 1 To NumShifts
  For k = 1 To MaxNumConfl
    If ConflictMatrix(i, k) <> 0 Then
      For j = 1 To NumEmp
        'Check if variable should be included
        'If preference is zero for either shift then employee is unavailable
        'and variable should not be in model
        If PriorityMatrix(i, j) = 0 Then EmpAvailable = False
        If PriorityMatrix(i, k) = 0 Then EmpAvailable = False
        'Skip writing the constraint to the file if EmpAvailable is false
        If EmpAvailable = True Then
          Print #1, "c" & ConstNum & ":"
          Print #1, "s" & i & "e" & j & " + s" & ConflictMatrix(i, k) & "e" & j & "
          <= 1"
          ConstNum = ConstNum + 1
        End If
        'Set availability to true
        EmpAvailable = True
      Next j
    End If
  Next k
Next i

```

```

'Write Set7 constraints
'Get maximum number of weekend shifts per month
Dim NumWeekendShifts, NumMonths, m As Integer
NumWeekendShifts = Sheets("Shifts").Cells(4, 8).Value
'j is loop for employees
For j = 1 To NumEmp
  Print #1, "c" & ConstNum & ":"
  'k is loop for weeks
  'm is loop for months
  NumMonths = Int(NumWeeks / 4)
  For m = 1 To NumMonths
    For k = 1 To 4
      'l is loop for days
      For l = 6 To 7
        If l = 6 Then DayWeekVarName = "em" & j & "w" & m * k & "d" & (m * k *
7) - 1
        If l = 7 Then DayWeekVarName = "em" & j & "w" & m * k & "d" & (m * k *
7)
        If k * l = 4 * 7 Then
          Print #1, DayWeekVarName & " <= " & NumWeekendShifts

```

```

        Else
            Print #1, DayWeekVarName & " + "
        End If
    Next l
Next k
Next m
ConstNum = ConstNum + 1
Next j

'Set all dummy variables to 0
Print #1, "dum1 = 0"
Print #1, "dum2 = 0"
Print #1, "dum3 = 0"

'Write Integer requirements
Print #1, "INTEGER"
For i = 1 To NumShifts
    For j = 1 To NumEmp
        'Check if variable should be included
        'If preference is zero for then employee is unavailable
        'and variable should not be in model
        If PriorityMatrix(i, j) = 0 Then EmpAvailable = False
        'Skip writing the variable to the file if EmpAvailable is false
        If EmpAvailable = True Then
            Print #1, "s" & i & "e" & j
        End If
        'Set availability to true
        EmpAvailable = True
    Next j
Next i

'Write END
Print #1, "END"
'Close mps file
Close #1

Exit Sub

ErrorHandler:
MsgBox ("Error")
'Close mps file
Close #1

```

End Sub

APPENDIX D

CHAPTER FOUR CPLEX OUTPUT

Log started (V9.1.0) Sun Jul 17 18:29:45 2005

Problem 'wle20.lp' read.
 Read time = 0.01 sec.
 Tried aggregator 1 time.
 LP Presolve eliminated 850 rows and 331 columns.
 Aggregator did 1 substitutions.
 Reduced LP has 1077 rows, 935 columns, and 3680 nonzeros.
 Presolve time = 0.01 sec.
 Initializing dual steep norms . . .

Iteration log . . .

| | | | | |
|------------|-----|----------------|---|---------------|
| Iteration: | 1 | Dual objective | = | 122110.500000 |
| Iteration: | 62 | Dual objective | = | 91007.000000 |
| Iteration: | 127 | Dual objective | = | 85149.500000 |
| Iteration: | 193 | Dual objective | = | 79992.000000 |

Dual simplex - Optimal: Objective = 7.9602500000e+04
 Solution time = 0.02 sec. Iterations = 237 (0)

| Variable Name | Solution Value |
|---------------|----------------|
| s1e10 | 1.000000 |
| s2e6 | 1.000000 |
| s3e1 | 1.000000 |
| s4e19 | 1.000000 |
| s5e3 | 1.000000 |
| s6e11 | 1.000000 |
| s7e5 | 1.000000 |
| s8e14 | 1.000000 |
| s9e9 | 1.000000 |
| s10e6 | 1.000000 |
| s11e10 | 1.000000 |
| s12e2 | 1.000000 |
| s13e5 | 1.000000 |
| s14e12 | 1.000000 |
| s15e3 | 1.000000 |
| s16e8 | 1.000000 |
| s17e20 | 1.000000 |
| s18e7 | 1.000000 |
| s19e10 | 1.000000 |
| s20e12 | 1.000000 |
| s21e9 | 1.000000 |
| s22e1 | 1.000000 |
| s23e7 | 1.000000 |
| s24e11 | 1.000000 |
| s25e4 | 1.000000 |
| s26e6 | 1.000000 |
| s27e17 | 1.000000 |
| s28e12 | 1.000000 |
| s29e13 | 1.000000 |
| s30e20 | 1.000000 |
| s31e1 | 1.000000 |
| s32e9 | 1.000000 |

| | |
|----------|----------|
| s33e18 | 1.000000 |
| s34e14 | 1.000000 |
| s35e2 | 1.000000 |
| s36e15 | 1.000000 |
| s37e5 | 1.000000 |
| s38e11 | 1.000000 |
| s39e10 | 1.000000 |
| s40e7 | 1.000000 |
| s41e8 | 1.000000 |
| s42e15 | 1.000000 |
| s43e3 | 1.000000 |
| s44e4 | 1.000000 |
| s45e1 | 1.000000 |
| s46e5 | 1.000000 |
| s47e8 | 1.000000 |
| s48e4 | 1.000000 |
| s49e2 | 1.000000 |
| s50e7 | 1.000000 |
| s51e16 | 1.000000 |
| s52e6 | 1.000000 |
| s53e18 | 1.000000 |
| s54e11 | 1.000000 |
| s55e13 | 1.000000 |
| s56e19 | 1.000000 |
| s57e16 | 1.000000 |
| s58e2 | 1.000000 |
| s59e9 | 1.000000 |
| s60e3 | 1.000000 |
| s61e4 | 1.000000 |
| s62e17 | 1.000000 |
| s63e8 | 1.000000 |
| em1w1d1 | 1.000000 |
| em3w1d1 | 1.000000 |
| em5w1d1 | 1.000000 |
| em6w1d1 | 1.000000 |
| em9w1d1 | 1.000000 |
| em10w1d1 | 1.000000 |
| em11w1d1 | 1.000000 |
| em14w1d1 | 1.000000 |
| em19w1d1 | 1.000000 |
| em2w1d2 | 1.000000 |
| em3w1d2 | 1.000000 |
| em5w1d2 | 1.000000 |
| em6w1d2 | 1.000000 |
| em7w1d2 | 1.000000 |
| em8w1d2 | 1.000000 |
| em10w1d2 | 1.000000 |
| em12w1d2 | 1.000000 |
| em20w1d2 | 1.000000 |
| em1w1d3 | 1.000000 |
| em4w1d3 | 1.000000 |
| em6w1d3 | 1.000000 |
| em7w1d3 | 1.000000 |
| em9w1d3 | 1.000000 |
| em10w1d3 | 1.000000 |

| | |
|----------|----------|
| em11wld3 | 1.000000 |
| em12wld3 | 1.000000 |
| em17wld3 | 1.000000 |
| em1wld4 | 1.000000 |
| em2wld4 | 1.000000 |
| em9wld4 | 1.000000 |
| em12wld4 | 1.000000 |
| em13wld4 | 1.000000 |
| em14wld4 | 1.000000 |
| em15wld4 | 1.000000 |
| em18wld4 | 1.000000 |
| em20wld4 | 1.000000 |
| em1wld5 | 1.000000 |
| em3wld5 | 1.000000 |
| em4wld5 | 1.000000 |
| em5wld5 | 1.000000 |
| em7wld5 | 1.000000 |
| em8wld5 | 1.000000 |
| em10wld5 | 1.000000 |
| em11wld5 | 1.000000 |
| em15wld5 | 1.000000 |
| em2wld6 | 1.000000 |
| em4wld6 | 1.000000 |
| em5wld6 | 1.000000 |
| em6wld6 | 1.000000 |
| em7wld6 | 1.000000 |
| em8wld6 | 1.000000 |
| em11wld6 | 1.000000 |
| em16wld6 | 1.000000 |
| em18wld6 | 1.000000 |
| em2wld7 | 1.000000 |
| em3wld7 | 1.000000 |
| em4wld7 | 1.000000 |
| em8wld7 | 1.000000 |
| em9wld7 | 1.000000 |
| em13wld7 | 1.000000 |
| em16wld7 | 1.000000 |
| em17wld7 | 1.000000 |
| em19wld7 | 1.000000 |

All other variables in the range 1-1267 are zero.

Problem 'wle20ip.lp' read.

Read time = 0.00 sec.

Tried aggregator 1 time.

MIP Presolve eliminated 850 rows and 331 columns.

Aggregator did 1 substitutions.

Reduced MIP has 1077 rows, 935 columns, and 3680 nonzeros.

Presolve time = 0.01 sec.

Clique table members: 1037

MIP emphasis: balance optimality and feasibility

Root relaxation solution time = 0.02 sec.

Integer optimal solution: Objective = 7.9602500000e+04

Solution time = 0.03 sec. Iterations = 322 Nodes = 0

Problem 'w2e20.lp' read.
 Read time = 0.01 sec.
 Tried aggregator 1 time.
 LP Presolve eliminated 1601 rows and 744 columns.
 Aggregator did 1 substitutions.
 Reduced LP has 2269 rows, 1753 columns, and 7434 nonzeros.
 Presolve time = 0.01 sec.
 Initializing dual steep norms . . .

Iteration log . . .
 Iteration: 1 Dual objective = 246189.000000
 Iteration: 63 Dual objective = 214922.000000
 Iteration: 163 Dual objective = 196781.500000
 Iteration: 262 Dual objective = 190105.750000
 Iteration: 361 Dual objective = 184657.500000
 Iteration: 449 Dual objective = 183576.750000

Dual simplex - Optimal: Objective = 1.8344200000e+05
 Solution time = 0.05 sec. Iterations = 513 (0)

Problem 'w2e20ip.lp' read.
 Read time = 0.01 sec.
 Tried aggregator 1 time.
 MIP Presolve eliminated 1601 rows and 744 columns.
 Aggregator did 1 substitutions.
 Reduced MIP has 2269 rows, 1753 columns, and 7434 nonzeros.
 Presolve time = 0.02 sec.
 Clique table members: 2229
 MIP emphasis: balance optimality and feasibility
 Root relaxation solution time = 0.06 sec.

| | Nodes | | Objective | IInf | Best Integer | Cuts/ Best Node | ItCnt |
|-------|-------|-------------|-------------------|-------|--------------|--------------------|-------|
| Node | Left | | Variable B Parent | Depth | | | |
| Gap | | | | | | | |
| 0 | 0 | 183442.0000 | | 71 | | 183442.0000 | 632 |
| | | 183409.5000 | | 18 | | Cuts: 89 | 696 |
| | | 183399.7500 | | 10 | | Cuts: 21 | 746 |
| * 0+ | 0 | | | 0 | 183363.0000 | 183399.7500 | 746 |
| 0.02% | | | | | | | |
| * | | 183396.5000 | | 0 | 183396.5000 | Cliques: 3 | 750 |
| 0.00% | | | | | | | |

Clique cuts applied: 24
 Gomory fractional cuts applied: 11

Integer optimal solution: Objective = 1.8339650000e+05
 Solution time = 0.17 sec. Iterations = 750 Nodes = 0

Log started (V9.1.0) Sun Jul 17 17:56:53 2005

Problem 'w12e20.lp' read.
 Read time = 0.10 sec.
 Tried aggregator 1 time.
 LP Presolve eliminated 9196 rows and 5318 columns.
 Aggregator did 4 substitutions.
 Reduced LP has 13738 rows, 9634 columns, and 43480 nonzeros.
 Presolve time = 0.06 sec.
 Initializing dual steep norms . . .

Iteration log . . .

| | | | | |
|------------|------|----------------|---|-----------------|
| Iteration: | 1 | Dual objective | = | 57069303.900000 |
| Iteration: | 358 | Dual objective | = | 52615722.200000 |
| Iteration: | 680 | Dual objective | = | 50621947.200000 |
| Iteration: | 993 | Dual objective | = | 49539532.850000 |
| Iteration: | 1306 | Dual objective | = | 48941165.000000 |
| Iteration: | 1585 | Dual objective | = | 48144221.100000 |
| Iteration: | 1892 | Dual objective | = | 47895102.950000 |
| Iteration: | 2152 | Dual objective | = | 47566514.575000 |
| Iteration: | 2412 | Dual objective | = | 47426345.412500 |
| Iteration: | 2659 | Dual objective | = | 47377069.943056 |
| Iteration: | 2882 | Dual objective | = | 47278573.700000 |
| Iteration: | 3101 | Dual objective | = | 47171073.833333 |
| Iteration: | 3328 | Dual objective | = | 47123625.433333 |
| Iteration: | 3519 | Dual objective | = | 47115675.873810 |
| Iteration: | 3705 | Dual objective | = | 47108769.452050 |

Dual simplex - Optimal: Objective = 4.7108769452e+07
 Solution time = 1.56 sec. Iterations = 3705 (0)

Problem 'w12e20ip.lp' read.
 Read time = 0.07 sec.
 Tried aggregator 1 time.
 MIP Presolve eliminated 10442 rows and 5339 columns.
 MIP Presolve modified 868 coefficients.
 Aggregator did 4 substitutions.
 Reduced MIP has 12492 rows, 9613 columns, and 43401 nonzeros.
 Presolve time = 0.16 sec.
 Clique table members: 12452
 MIP emphasis: balance optimality and feasibility
 Root relaxation solution time = 1.21 sec.

| Nodes | | Objective Variable B Parent | IInf Depth | Best Integer | Cuts/ | ItCnt |
|-------|------|--------------------------------|---------------|--------------|-------------|-------|
| Node | Left | | | | Best Node | |
| Gap | | | | | | |
| | 0 | 4.7099e+07 | 960 | | 4.7099e+07 | 4062 |
| | | 4.7076e+07 | 763 | | Cuts: 363 | 4707 |
| * | 0+ | | 0 | 4.7008e+07 | 4.7076e+07 | 4707 |
| 0.14% | | 4.7073e+07 | 435 | 4.7008e+07 | Cuts: 252 | 4947 |
| 0.14% | | 4.7072e+07 | 372 | 4.7008e+07 | Cliques: 14 | 5007 |
| 0.14% | | | | | | |

```

*      0+      0      0      4.7018e+07      4.7072e+07      5007
0.12%
*      0+      0      0      4.7054e+07      4.7072e+07      5007
0.04%
*      0+      0      0      4.7067e+07      4.7072e+07      5007
0.01%

```

Elapsed time = 10.12 sec. (tree size = 0.00 MB)

Clique cuts applied: 208

Gomory fractional cuts applied: 45

Integer optimal, tolerance (0.0001/1e-06) : Objective =

4.7067485000e+07

Current MIP best bound = 4.7072176250e+07 (gap = 4691.25, 0.01%)

Solution time = 10.87 sec. Iterations = 5072 Nodes = 2 (3)

Problem 'w4e20.lp' read.

Read time = 0.03 sec.

Tried aggregator 1 time.

LP Presolve eliminated 3261 rows and 1705 columns.

Aggregator did 1 substitutions.

Reduced LP has 4573 rows, 3300 columns, and 14604 nonzeros.

Presolve time = 0.01 sec.

Initializing dual steep norms . . .

Iteration log . . .

```

Iteration:      1      Dual objective      =      6323675.000000
Iteration:     88      Dual objective      =      5879590.000000
Iteration:    219      Dual objective      =      5549335.000000
Iteration:    372      Dual objective      =      5385128.333333
Iteration:    507      Dual objective      =      5239922.500000
Iteration:    656      Dual objective      =      5121212.500000
Iteration:    773      Dual objective      =      5105440.833333
Iteration:    898      Dual objective      =      5060940.833333
Iteration:   1010      Dual objective      =      5055779.953704
Iteration:   1110      Dual objective      =      5040283.888889
Iteration:   1207      Dual objective      =      5032525.608974

```

Dual simplex - Optimal: Objective = 5.0306182417e+06

Solution time = 0.22 sec. Iterations = 1294 (0)

Problem 'w4e20ip.lp' read.

Read time = 0.02 sec.

Tried aggregator 1 time.

MIP Presolve eliminated 3261 rows and 1705 columns.

Aggregator did 1 substitutions.

Reduced MIP has 4573 rows, 3300 columns, and 14604 nonzeros.

Presolve time = 0.04 sec.

Clique table members: 4533

MIP emphasis: balance optimality and feasibility

Root relaxation solution time = 0.13 sec.

Nodes

Cuts/

| Node Gap | Left | Objective Variable B Parent | IInf Depth | Best Integer | Best Node | ItCnt |
|----------|------|-----------------------------|------------|--------------|--------------|-------|
| 0 | 0 | 5030618.2417 | 327 | | 5030618.2417 | 1155 |
| | | 5027005.3777 | 271 | | Cuts: 269 | 1424 |
| | | 5026072.3333 | 137 | | Cuts: 232 | 1559 |
| | | 5025847.6250 | 152 | | Cliques: 11 | 1604 |
| | | 5025731.6667 | 143 | | Cliques: 7 | 1644 |
| * 0.06% | 0+ | 0 | 0 | 5022965.0000 | 5025731.6667 | 1644 |
| * 0.04% | 0+ | 0 | 0 | 5023770.0000 | 5025731.6667 | 1644 |
| * 0.03% | 0+ | 0 | 0 | 5024045.0000 | 5025731.6667 | 1644 |
| * 0.01% | 4 | 4 | 0 | 5025100.0000 | 5025690.8333 | 1700 |
| * 0.01% | 5 | 3 | 0 | 5025225.0000 | 5025690.8333 | 1707 |
| | | s242e2 D | 3 | 4 | | |
| | | s242e2 U | 3 | 4 | | |

Clique cuts applied: 107
Gomory fractional cuts applied: 43

Integer optimal, tolerance (0.0001/1e-06) : Objective =
5.0252250000e+06
Current MIP best bound = 5.0256908333e+06 (gap = 465.833, 0.01%)
Solution time = 2.14 sec. Iterations = 1707 Nodes = 6 (3)

Problem 'w52e20.lp' read.
Read time = 0.39 sec.
Tried aggregator 1 time.
LP Presolve eliminated 42798 rows and 25459 columns.
Aggregator did 16 substitutions.
Reduced LP has 59089 rows, 39493 columns, and 183643 nonzeros.
Presolve time = 0.30 sec.
Initializing dual steep norms . . .

Iteration log . . .

| | | | | |
|------------|-------|----------------|---|-------------------|
| Iteration: | 1 | Dual objective | = | 1079294544.900000 |
| Iteration: | 732 | Dual objective | = | 1026452718.700001 |
| Iteration: | 1208 | Dual objective | = | 1007896717.350001 |
| Iteration: | 1875 | Dual objective | = | 986069005.450001 |
| Iteration: | 2526 | Dual objective | = | 964575678.600002 |
| Iteration: | 3201 | Dual objective | = | 954946214.700002 |
| Iteration: | 3809 | Dual objective | = | 946444003.000003 |
| Iteration: | 4488 | Dual objective | = | 932469531.550003 |
| Iteration: | 5423 | Dual objective | = | 921571715.100002 |
| Iteration: | 6138 | Dual objective | = | 916259411.625003 |
| Iteration: | 6853 | Dual objective | = | 908584913.466671 |
| Iteration: | 7526 | Dual objective | = | 906427589.795838 |
| Iteration: | 8233 | Dual objective | = | 904335752.500005 |
| Iteration: | 8917 | Dual objective | = | 903488376.490282 |
| Iteration: | 9521 | Dual objective | = | 902800464.221359 |
| Iteration: | 10139 | Dual objective | = | 902357825.588542 |
| Iteration: | 10733 | Dual objective | = | 900171320.041204 |

```

Iteration: 11365   Dual objective   =      897194064.482788
Iteration: 11992   Dual objective   =      896545640.482142
Iteration: 12612   Dual objective   =      896085408.583471
Iteration: 13229   Dual objective   =      895853503.492938
Iteration: 13786   Dual objective   =      895655075.838842
Iteration: 14333   Dual objective   =      895541668.713304
Iteration: 14859   Dual objective   =      895477957.820736
Iteration: 15402   Dual objective   =      895404683.273381
Iteration: 15821   Dual objective   =      894703935.102226
Iteration: 16361   Dual objective   =      894213914.280914
Iteration: 16894   Dual objective   =      893933458.040760

```

```

Dual simplex - Optimal: Objective =      8.9392888657e+08
Solution time =      7.64 sec.  Iterations = 17054 (0)

```

```

Problem 'w52e20ip.lp' read.
Read time =      0.43 sec.
Tried aggregator 1 time.
Presolve has eliminated 61033 rows and 27649 columns...
Aggregator has done 16 substitutions...
MIP Presolve eliminated 61033 rows and 27649 columns.
MIP Presolve modified 10959 coefficients.
Aggregator did 16 substitutions.
Reduced MIP has 40854 rows, 37303 columns, and 172474 nonzeros.
Presolve time =      5.12 sec.
Clique table members: 40784
MIP emphasis: balance optimality and feasibility
Root relaxation solution time =      7.42 sec.

```

| | Nodes | | | | Cuts/ | |
|------|-------|-------------------|-------|--------------|-------------|-------|
| Node | Left | Objective | IInf | Best Integer | Best Node | ItCnt |
| Gap | | Variable B Parent | Depth | | | |
| 0 | 0 | 8.8556e+08 | 4723 | | 8.8556e+08 | 14999 |
| | | 8.8510e+08 | 3317 | | Cuts: 700 | 17496 |
| | | 8.8502e+08 | 2379 | | Cuts: 395 | 18476 |
| | | 8.8500e+08 | 2457 | | Cliques: 66 | 18778 |

Heuristic still looking

Heuristic complete

Elapsed time = 73.59 sec. (tree size = 0.00 MB)

| | | | | | | | |
|-------|-----|-----------|------------|------|------------|------------|-------|
| * | 70+ | 70 | | 0 | 8.8479e+08 | 8.8500e+08 | 20269 |
| 0.02% | | | | | | | |
| * | 80+ | 80 | | 0 | 8.8481e+08 | 8.8500e+08 | 20581 |
| 0.02% | | | | | | | |
| * | 90+ | 90 | | 0 | 8.8482e+08 | 8.8500e+08 | 20800 |
| 0.02% | | | | | | | |
| * | 90+ | 90 | | 0 | 8.8484e+08 | 8.8500e+08 | 20800 |
| 0.02% | | | | | | | |
| * | 94 | 90 | | 0 | 8.8487e+08 | 8.8500e+08 | 20852 |
| 0.01% | | s1958e5 U | 93 | 74 | | | |
| | 100 | 92 | 8.8498e+08 | 2378 | 8.8487e+08 | 8.8500e+08 | 21108 |
| 0.01% | | s2117e4 U | 99 | 4 | | | |
| | 200 | 192 | 8.8489e+08 | 77 | 8.8487e+08 | 8.8500e+08 | 23251 |
| 0.01% | | s503e2 U | 199 | 83 | | | |

```

*   206   197           0   8.8487e+08   8.8500e+08   23418
0.01%           s532e10 U   205   88
   300   285   8.8491e+08   575   8.8487e+08   8.8500e+08   24822
0.01%           s1669e3 U   299   92
   400   381   8.8489e+08   589   8.8487e+08   8.8500e+08   26540
0.01%           s1759e10 U   399   87
   500   477   8.8490e+08   790   8.8487e+08   8.8500e+08   28618
0.01%           s1711e3 D   499   64
   600   573   8.8492e+08   838   8.8487e+08   8.8499e+08   30343
0.01%           s2559e2 U   599   56
   700   669   8.8494e+08   1199   8.8487e+08   8.8499e+08   32030
0.01%           s2545e1 U   699   58
   800   766   8.8495e+08   1454   8.8487e+08   8.8499e+08   33665
0.01%           s2337e6 D   799   47
*   860   801           0   8.8488e+08   8.8499e+08   34397
0.01%           s2835e7 U   859   107
*   863   799           0   8.8488e+08   8.8499e+08   34459
0.01%           s2835e4 D   861   108
   900   834   8.8495e+08   1680   8.8488e+08   8.8499e+08   35461
0.01%           s2529e3 U   899   33
*   960+   813           0   8.8490e+08   8.8499e+08   36195
0.01%
   1000   848   8.8495e+08   1660   8.8490e+08   8.8499e+08   36945
0.01%           s1692e7 U   999   37
Elapsed time = 203.34 sec. (tree size = 15.30 MB)
*  1000+   722           0   8.8491e+08   8.8499e+08   36945
0.01%

```

Clique cuts applied: 713
 Gomory fractional cuts applied: 70

Integer optimal, tolerance (0.0001/1e-06) : Objective =
 8.8491464500e+08
 Current MIP best bound = 8.8499398452e+08 (gap = 79339.5, 0.01%)
 Solution time = 204.15 sec. Iterations = 36945 Nodes = 1000 (723)

APPENDIX E

CHAPTER FIVE VISUAL BASIC CODE

```
Private Sub CommandButton4_Click()
```

```
'Will provide the preferences for rerostering
```

```
'Shane Knighton 7-28-05
```

```
'Find optimal assignments and increase their preferences by BigM
```

```
Dim BigM, i As Integer
```

```
BigM = 10000
```

```
Dim NumShifts As Integer
```

```
NumShifts = Sheets("Shifts").Cells(2, 6).Value
```

```
'Get optimal shifts and employee combination
```

```
Dim Shift, Emp As Integer
```

```
'i is loop for shifts
```

```
For i = 1 To NumShifts
```

```
    Shift = Mid(Sheets("Employees").Cells(9 + i, 60).Value, 2, 1)
```

```
    EmpNumStart = 4
```

```
    If Mid(Sheets("Employees").Cells(9 + i, 60).Value, 3, 1) <> "e" Then
```

```
        Shift = Shift & Mid(Sheets("Employees").Cells(9 + i, 60).Value, 3, 1)
```

```
        EmpNumStart = 5
```

```
        If Mid(Sheets("Employees").Cells(9 + i, 60).Value, 4, 1) <> "e" Then
```

```
            Shift = Shift & Mid(Sheets("Employees").Cells(9 + i, 60).Value, 4, 1)
```

```
            EmpNumStart = 6
```

```
        End If
```

```
    End If
```

```
    Emp = Mid(Sheets("Employees").Cells(9 + i, 60).Value, EmpNumStart, 1)
```

```
    If Mid(Sheets("Employees").Cells(9 + i, 60).Value, EmpNumStart + 1, 1) <> ""
```

```
Then
```

```
    Emp = Emp & Mid(Sheets("Employees").Cells(9 + i, 60).Value,
```

```
EmpNumStart + 1, 1)
```

```
End If
```

```
'Check
```

```
'Sheets("Employees").Cells(9 + i, 59).Value = Shift & " " & Emp
```

```
'Add Big M to preferences
```

```
Dim OldPref As Double
```

```
OldPref = Sheets("Employees").Cells(9 + Shift, 3 + Emp).Value
```

```
Sheets("Employees").Cells(9 + Shift, 3 + Emp).Value = OldPref + BigM
```

```
Sheets("Employees").Select
```

```
Range(Cells(9 + Shift, 3 + Emp), Cells(9 + Shift, 3 + Emp)).Select
```

```
With Selection.Interior
```

```
    .ColorIndex = 4
```

```

        .Pattern = xlSolid
    End With

Next i

End Sub

Private Sub CommandButton5_Click()

'Will show the rerostered schedule rerostering
'Shane Knighton 7-28-05

'Get optimal shifts and employee combination
Dim Shift, Emp, i As Integer
Dim NumShifts As Integer
NumShifts = Sheets("Shifts").Cells(2, 6).Value

'i is loop for shifts
For i = 1 To NumShifts
    Shift = Mid(Sheets("Employees").Cells(9 + i, 65).Value, 2, 1)
    EmpNumStart = 4
    If Mid(Sheets("Employees").Cells(9 + i, 65).Value, 3, 1) <> "e" Then
        Shift = Shift & Mid(Sheets("Employees").Cells(9 + i, 65).Value, 3, 1)
        EmpNumStart = 5
        If Mid(Sheets("Employees").Cells(9 + i, 65).Value, 4, 1) <> "e" Then
            Shift = Shift & Mid(Sheets("Employees").Cells(9 + i, 65).Value, 4, 1)
            EmpNumStart = 6
        End If
    End If
    Emp = Mid(Sheets("Employees").Cells(9 + i, 65).Value, EmpNumStart, 1)
    If Mid(Sheets("Employees").Cells(9 + i, 65).Value, EmpNumStart + 1, 1) <> ""
Then
        Emp = Emp & Mid(Sheets("Employees").Cells(9 + i, 65).Value,
EmpNumStart + 1, 1)
    End If

'Check
'Sheets("Employees").Cells(9 + i, 59).Value = Shift & " " & Emp

'Show new optimal

Sheets("Employees").Select
Range(Cells(9 + Shift, 3 + Emp), Cells(9 + Shift, 3 + Emp)).Select
    Selection.Font.Bold = True

```



```
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With

Next i

End Sub
```

APPENDIX F

CHAPTER FIVE CPLEX OUTPUT

Log started (V9.1.0) Thu Jul 28 13:04:04 2005

Problem 'wle20.lp' read.
 Read time = 0.01 sec.
 Tried aggregator 1 time.
 LP Presolve eliminated 905 rows and 343 columns.
 Aggregator did 1 substitutions.
 Reduced LP has 1079 rows, 942 columns, and 3698 nonzeros.
 Presolve time = -0.00 sec.
 Initializing dual steep norms . . .

Iteration log . . .

| | | | | |
|------------|-----|----------------|---|---------------|
| Iteration: | 1 | Dual objective | = | 234369.000000 |
| Iteration: | 62 | Dual objective | = | 176019.000000 |
| Iteration: | 125 | Dual objective | = | 163800.000000 |
| Iteration: | 193 | Dual objective | = | 155394.000000 |

Dual simplex - Optimal: Objective = 1.5458100000e+05
 Solution time = 0.02 sec. Iterations = 247 (0)

| Variable Name | Solution Value |
|---------------|----------------|
| s1e12 | 1.000000 |
| s2e5 | 1.000000 |
| s3e3 | 1.000000 |
| s4e1 | 1.000000 |
| s5e20 | 1.000000 |
| s6e4 | 1.000000 |
| s7e6 | 1.000000 |
| s8e11 | 1.000000 |
| s9e14 | 1.000000 |
| s10e18 | 1.000000 |
| s11e14 | 1.000000 |
| s12e8 | 1.000000 |
| s13e17 | 1.000000 |
| s14e12 | 1.000000 |
| s15e2 | 1.000000 |
| s16e7 | 1.000000 |
| s17e4 | 1.000000 |
| s18e1 | 1.000000 |
| s19e9 | 1.000000 |
| s20e7 | 1.000000 |
| s21e6 | 1.000000 |
| s22e2 | 1.000000 |
| s23e10 | 1.000000 |
| s24e4 | 1.000000 |
| s25e15 | 1.000000 |
| s26e1 | 1.000000 |
| s27e3 | 1.000000 |
| s28e11 | 1.000000 |
| s29e7 | 1.000000 |
| s30e16 | 1.000000 |
| s31e13 | 1.000000 |
| s32e4 | 1.000000 |

| | |
|----------|----------|
| s33e10 | 1.000000 |
| s34e3 | 1.000000 |
| s35e19 | 1.000000 |
| s36e15 | 1.000000 |
| s37e16 | 1.000000 |
| s38e11 | 1.000000 |
| s39e2 | 1.000000 |
| s40e5 | 1.000000 |
| s41e10 | 1.000000 |
| s42e9 | 1.000000 |
| s43e13 | 1.000000 |
| s44e6 | 1.000000 |
| s45e8 | 1.000000 |
| s46e3 | 1.000000 |
| s47e19 | 1.000000 |
| s48e8 | 1.000000 |
| s49e10 | 1.000000 |
| s50e9 | 1.000000 |
| s51e2 | 1.000000 |
| s52e20 | 1.000000 |
| s53e5 | 1.000000 |
| s54e17 | 1.000000 |
| s55e18 | 1.000000 |
| s56e8 | 1.000000 |
| s57e1 | 1.000000 |
| s58e9 | 1.000000 |
| s59e6 | 1.000000 |
| s60e12 | 1.000000 |
| s61e11 | 1.000000 |
| s62e7 | 1.000000 |
| s63e5 | 1.000000 |
| em1w1d1 | 1.000000 |
| em3w1d1 | 1.000000 |
| em4w1d1 | 1.000000 |
| em5w1d1 | 1.000000 |
| em6w1d1 | 1.000000 |
| em11w1d1 | 1.000000 |
| em12w1d1 | 1.000000 |
| em14w1d1 | 1.000000 |
| em20w1d1 | 1.000000 |
| em1w1d2 | 1.000000 |
| em2w1d2 | 1.000000 |
| em4w1d2 | 1.000000 |
| em7w1d2 | 1.000000 |
| em8w1d2 | 1.000000 |
| em12w1d2 | 1.000000 |
| em14w1d2 | 1.000000 |
| em17w1d2 | 1.000000 |
| em18w1d2 | 1.000000 |
| em1w1d3 | 1.000000 |
| em2w1d3 | 1.000000 |
| em3w1d3 | 1.000000 |
| em4w1d3 | 1.000000 |
| em6w1d3 | 1.000000 |
| em7w1d3 | 1.000000 |

| | |
|----------|----------|
| em9wld3 | 1.000000 |
| em10wld3 | 1.000000 |
| em15wld3 | 1.000000 |
| em3wld4 | 1.000000 |
| em4wld4 | 1.000000 |
| em7wld4 | 1.000000 |
| em10wld4 | 1.000000 |
| em11wld4 | 1.000000 |
| em13wld4 | 1.000000 |
| em15wld4 | 1.000000 |
| em16wld4 | 1.000000 |
| em19wld4 | 1.000000 |
| em2wld5 | 1.000000 |
| em5wld5 | 1.000000 |
| em6wld5 | 1.000000 |
| em8wld5 | 1.000000 |
| em9wld5 | 1.000000 |
| em10wld5 | 1.000000 |
| em11wld5 | 1.000000 |
| em13wld5 | 1.000000 |
| em16wld5 | 1.000000 |
| em2wld6 | 1.000000 |
| em3wld6 | 1.000000 |
| em5wld6 | 1.000000 |
| em8wld6 | 1.000000 |
| em9wld6 | 1.000000 |
| em10wld6 | 1.000000 |
| em17wld6 | 1.000000 |
| em19wld6 | 1.000000 |
| em20wld6 | 1.000000 |
| em1wld7 | 1.000000 |
| em5wld7 | 1.000000 |
| em6wld7 | 1.000000 |
| em7wld7 | 1.000000 |
| em8wld7 | 1.000000 |
| em9wld7 | 1.000000 |
| em11wld7 | 1.000000 |
| em12wld7 | 1.000000 |
| em18wld7 | 1.000000 |

All other variables in the range 1-1286 are zero.

Log started (V9.1.0) Thu Jul 28 13:59:56 2005

Problem 'wle20s1.lp' read.
 Read time = 0.01 sec.
 Tried aggregator 1 time.
 LP Presolve eliminated 905 rows and 343 columns.
 Aggregator did 1 substitutions.
 Reduced LP has 1079 rows, 941 columns, and 3696 nonzeros.
 Presolve time = -0.00 sec.
 Initializing dual steep norms . . .

Iteration log . . .

Iteration: 1 Dual objective = 775949.000000

Dual simplex - Optimal: Objective = 7.74329000000e+05

Solution time = 0.01 sec. Iterations = 5 (0)

| Variable Name | Solution Value |
|---------------|----------------|
| s1e13 | 1.000000 |
| s2e5 | 1.000000 |
| s3e3 | 1.000000 |
| s4e1 | 1.000000 |
| s5e20 | 1.000000 |
| s6e4 | 1.000000 |
| s7e6 | 1.000000 |
| s8e11 | 1.000000 |
| s9e14 | 1.000000 |
| s10e18 | 1.000000 |
| s11e14 | 1.000000 |
| s12e8 | 1.000000 |
| s13e17 | 1.000000 |
| s14e12 | 1.000000 |
| s15e2 | 1.000000 |
| s16e7 | 1.000000 |
| s17e4 | 1.000000 |
| s18e1 | 1.000000 |
| s19e9 | 1.000000 |
| s20e7 | 1.000000 |
| s21e6 | 1.000000 |
| s22e2 | 1.000000 |
| s23e10 | 1.000000 |
| s24e4 | 1.000000 |
| s25e15 | 1.000000 |
| s26e1 | 1.000000 |
| s27e3 | 1.000000 |
| s28e11 | 1.000000 |
| s29e7 | 1.000000 |
| s30e16 | 1.000000 |
| s31e13 | 1.000000 |
| s32e4 | 1.000000 |
| s33e10 | 1.000000 |
| s34e3 | 1.000000 |
| s35e19 | 1.000000 |
| s36e15 | 1.000000 |
| s37e16 | 1.000000 |
| s38e11 | 1.000000 |
| s39e2 | 1.000000 |
| s40e5 | 1.000000 |
| s41e10 | 1.000000 |
| s42e9 | 1.000000 |
| s43e13 | 1.000000 |
| s44e6 | 1.000000 |
| s45e8 | 1.000000 |
| s46e3 | 1.000000 |
| s47e19 | 1.000000 |
| s48e8 | 1.000000 |

| | |
|----------|----------|
| s49e10 | 1.000000 |
| s50e9 | 1.000000 |
| s51e2 | 1.000000 |
| s52e20 | 1.000000 |
| s53e5 | 1.000000 |
| s54e17 | 1.000000 |
| s55e18 | 1.000000 |
| s56e8 | 1.000000 |
| s57e1 | 1.000000 |
| s58e9 | 1.000000 |
| s59e6 | 1.000000 |
| s60e12 | 1.000000 |
| s61e11 | 1.000000 |
| s62e7 | 1.000000 |
| s63e5 | 1.000000 |
| em1w1d1 | 1.000000 |
| em3w1d1 | 1.000000 |
| em4w1d1 | 1.000000 |
| em5w1d1 | 1.000000 |
| em6w1d1 | 1.000000 |
| em11w1d1 | 1.000000 |
| em13w1d1 | 1.000000 |
| em14w1d1 | 1.000000 |
| em20w1d1 | 1.000000 |
| em1w1d2 | 1.000000 |
| em2w1d2 | 1.000000 |
| em4w1d2 | 1.000000 |
| em7w1d2 | 1.000000 |
| em8w1d2 | 1.000000 |
| em12w1d2 | 1.000000 |
| em14w1d2 | 1.000000 |
| em17w1d2 | 1.000000 |
| em18w1d2 | 1.000000 |
| em1w1d3 | 1.000000 |
| em2w1d3 | 1.000000 |
| em3w1d3 | 1.000000 |
| em4w1d3 | 1.000000 |
| em6w1d3 | 1.000000 |
| em7w1d3 | 1.000000 |
| em9w1d3 | 1.000000 |
| em10w1d3 | 1.000000 |
| em15w1d3 | 1.000000 |
| em3w1d4 | 1.000000 |
| em4w1d4 | 1.000000 |
| em7w1d4 | 1.000000 |
| em10w1d4 | 1.000000 |
| em11w1d4 | 1.000000 |
| em13w1d4 | 1.000000 |
| em15w1d4 | 1.000000 |
| em16w1d4 | 1.000000 |
| em19w1d4 | 1.000000 |
| em2w1d5 | 1.000000 |
| em5w1d5 | 1.000000 |
| em6w1d5 | 1.000000 |
| em8w1d5 | 1.000000 |

| | |
|----------|----------|
| em9wld5 | 1.000000 |
| em10wld5 | 1.000000 |
| em11wld5 | 1.000000 |
| em13wld5 | 1.000000 |
| em16wld5 | 1.000000 |
| em2wld6 | 1.000000 |
| em3wld6 | 1.000000 |
| em5wld6 | 1.000000 |
| em8wld6 | 1.000000 |
| em9wld6 | 1.000000 |
| em10wld6 | 1.000000 |
| em17wld6 | 1.000000 |
| em19wld6 | 1.000000 |
| em20wld6 | 1.000000 |
| em1wld7 | 1.000000 |
| em5wld7 | 1.000000 |
| em6wld7 | 1.000000 |
| em7wld7 | 1.000000 |
| em8wld7 | 1.000000 |
| em9wld7 | 1.000000 |
| em11wld7 | 1.000000 |
| em12wld7 | 1.000000 |
| em18wld7 | 1.000000 |

All other variables in the range 1-1285 are zero.

Log started (V9.1.0) Thu Jul 28 14:28:25 2005

Problem 'wle20s2.lp' read.
 Read time = 0.01 sec.
 Tried aggregator 1 time.
 LP Presolve eliminated 914 rows and 365 columns.
 Aggregator did 1 substitutions.
 Reduced LP has 1004 rows, 871 columns, and 3417 nonzeros.
 Presolve time = 0.01 sec.
 Initializing dual steep norms . . .

Iteration log . . .
 Iteration: 1 Dual objective = 742375.000000

Dual simplex - Optimal: Objective = 7.3161400000e+05
 Solution time = 0.01 sec. Iterations = 34 (0)

| Variable Name | Solution Value |
|---------------|----------------|
| s1e15 | 1.000000 |
| s2e5 | 1.000000 |
| s3e3 | 1.000000 |
| s4e1 | 1.000000 |
| s5e20 | 1.000000 |
| s6e4 | 1.000000 |
| s7e6 | 1.000000 |
| s8e11 | 1.000000 |
| s9e14 | 1.000000 |
| s10e18 | 1.000000 |

| | |
|---------|----------|
| s11e14 | 1.000000 |
| s12e8 | 1.000000 |
| s13e17 | 1.000000 |
| s14e13 | 1.000000 |
| s15e2 | 1.000000 |
| s16e7 | 1.000000 |
| s17e4 | 1.000000 |
| s18e1 | 1.000000 |
| s19e9 | 1.000000 |
| s20e7 | 1.000000 |
| s21e6 | 1.000000 |
| s22e2 | 1.000000 |
| s23e10 | 1.000000 |
| s24e4 | 1.000000 |
| s25e15 | 1.000000 |
| s26e1 | 1.000000 |
| s27e3 | 1.000000 |
| s28e17 | 1.000000 |
| s29e7 | 1.000000 |
| s30e16 | 1.000000 |
| s31e13 | 1.000000 |
| s32e4 | 1.000000 |
| s33e10 | 1.000000 |
| s34e3 | 1.000000 |
| s35e19 | 1.000000 |
| s36e15 | 1.000000 |
| s37e16 | 1.000000 |
| s38e14 | 1.000000 |
| s39e2 | 1.000000 |
| s40e5 | 1.000000 |
| s41e10 | 1.000000 |
| s42e9 | 1.000000 |
| s43e13 | 1.000000 |
| s44e6 | 1.000000 |
| s45e8 | 1.000000 |
| s46e3 | 1.000000 |
| s47e19 | 1.000000 |
| s48e8 | 1.000000 |
| s49e10 | 1.000000 |
| s50e9 | 1.000000 |
| s51e2 | 1.000000 |
| s52e20 | 1.000000 |
| s53e5 | 1.000000 |
| s54e17 | 1.000000 |
| s55e18 | 1.000000 |
| s56e8 | 1.000000 |
| s57e1 | 1.000000 |
| s58e9 | 1.000000 |
| s59e6 | 1.000000 |
| s60e13 | 1.000000 |
| s61e11 | 1.000000 |
| s62e7 | 1.000000 |
| s63e5 | 1.000000 |
| em1w1d1 | 1.000000 |
| em3w1d1 | 1.000000 |

| | |
|----------|----------|
| em4w1d1 | 1.000000 |
| em5w1d1 | 1.000000 |
| em6w1d1 | 1.000000 |
| em11w1d1 | 1.000000 |
| em14w1d1 | 1.000000 |
| em15w1d1 | 1.000000 |
| em20w1d1 | 1.000000 |
| em1w1d2 | 1.000000 |
| em2w1d2 | 1.000000 |
| em4w1d2 | 1.000000 |
| em7w1d2 | 1.000000 |
| em8w1d2 | 1.000000 |
| em13w1d2 | 1.000000 |
| em14w1d2 | 1.000000 |
| em17w1d2 | 1.000000 |
| em18w1d2 | 1.000000 |
| em1w1d3 | 1.000000 |
| em2w1d3 | 1.000000 |
| em3w1d3 | 1.000000 |
| em4w1d3 | 1.000000 |
| em6w1d3 | 1.000000 |
| em7w1d3 | 1.000000 |
| em9w1d3 | 1.000000 |
| em10w1d3 | 1.000000 |
| em15w1d3 | 1.000000 |
| em3w1d4 | 1.000000 |
| em4w1d4 | 1.000000 |
| em7w1d4 | 1.000000 |
| em10w1d4 | 1.000000 |
| em13w1d4 | 1.000000 |
| em15w1d4 | 1.000000 |
| em16w1d4 | 1.000000 |
| em17w1d4 | 1.000000 |
| em19w1d4 | 1.000000 |
| em2w1d5 | 1.000000 |
| em5w1d5 | 1.000000 |
| em6w1d5 | 1.000000 |
| em8w1d5 | 1.000000 |
| em9w1d5 | 1.000000 |
| em10w1d5 | 1.000000 |
| em13w1d5 | 1.000000 |
| em14w1d5 | 1.000000 |
| em16w1d5 | 1.000000 |
| em2w1d6 | 1.000000 |
| em3w1d6 | 1.000000 |
| em5w1d6 | 1.000000 |
| em8w1d6 | 1.000000 |
| em9w1d6 | 1.000000 |
| em10w1d6 | 1.000000 |
| em17w1d6 | 1.000000 |
| em19w1d6 | 1.000000 |
| em20w1d6 | 1.000000 |
| em1w1d7 | 1.000000 |
| em5w1d7 | 1.000000 |
| em6w1d7 | 1.000000 |

```

em7wld7          1.000000
em8wld7          1.000000
em9wld7          1.000000
em11wld7         1.000000
em13wld7         1.000000
em18wld7         1.000000
em12wld3         1.000000
em12wld4         1.000000
em12wld5         1.000000
em12wld6         1.000000
All other variables in the range 1-1237 are zero.

```

Log started (V9.1.0) Thu Jul 28 14:52:44 2005

```

Problem 'wle20s3.lp' read.
Read time =      0.01 sec.
Tried aggregator 1 time.
MIP Presolve eliminated 801 rows and 346 columns.
Aggregator did 1 substitutions.
Reduced MIP has 810 rows, 687 columns, and 2700 nonzeros.
Presolve time =      0.01 sec.
Clique table members: 781
MIP emphasis: balance optimality and feasibility
Root relaxation solution time =      0.01 sec.

```

| | Nodes | | Objective | IInf | Best Integer | Cuts/ Best Node | ItCnt |
|-------|-------|------|-------------------|-------|--------------|--------------------|-------|
| Gap | Node | Left | Variable B Parent | Depth | | | |
| | 0 | 0 | 604668.5000 | 16 | | 604668.5000 | 103 |
| * | 0+ | 0 | | 0 | 604598.0000 | 604668.5000 | 103 |
| 0.01% | | | | | | | |
| * | | | 604658.0000 | 0 | 604658.0000 | Fract: 16 | 112 |
| 0.00% | | | | | | | |

Gomory fractional cuts applied: 1

```

Integer optimal solution: Objective =      6.0465800003e+05
Solution time =      0.03 sec.  Iterations = 112  Nodes = 0

```

| Variable Name | Solution Value |
|---------------|----------------|
| s1e16 | 1.000000 |
| s2e5 | 1.000000 |
| s3e3 | 1.000000 |
| s4e1 | 1.000000 |
| s5e20 | 1.000000 |
| s6e4 | 1.000000 |
| s7e6 | 1.000000 |
| s8e18 | 1.000000 |
| s9e14 | 1.000000 |
| s10e18 | 1.000000 |
| s11e14 | 1.000000 |
| s12e8 | 1.000000 |

| | |
|---------|----------|
| s13e17 | 1.000000 |
| s14e20 | 1.000000 |
| s15e2 | 1.000000 |
| s16e7 | 1.000000 |
| s17e4 | 1.000000 |
| s18e1 | 1.000000 |
| s19e9 | 1.000000 |
| s20e7 | 1.000000 |
| s21e6 | 1.000000 |
| s22e2 | 1.000000 |
| s23e19 | 1.000000 |
| s24e4 | 1.000000 |
| s25e15 | 1.000000 |
| s26e1 | 1.000000 |
| s27e3 | 1.000000 |
| s28e17 | 1.000000 |
| s29e7 | 1.000000 |
| s30e16 | 1.000000 |
| s31e9 | 1.000000 |
| s32e4 | 1.000000 |
| s33e10 | 1.000000 |
| s34e3 | 1.000000 |
| s35e19 | 1.000000 |
| s36e15 | 1.000000 |
| s37e16 | 1.000000 |
| s38e17 | 1.000000 |
| s39e2 | 1.000000 |
| s40e5 | 1.000000 |
| s41e18 | 1.000000 |
| s42e9 | 1.000000 |
| s43e19 | 1.000000 |
| s44e6 | 1.000000 |
| s45e8 | 1.000000 |
| s46e3 | 1.000000 |
| s47e19 | 1.000000 |
| s48e8 | 1.000000 |
| s49e15 | 1.000000 |
| s50e9 | 1.000000 |
| s51e2 | 1.000000 |
| s52e20 | 1.000000 |
| s53e5 | 1.000000 |
| s54e17 | 1.000000 |
| s55e18 | 1.000000 |
| s56e8 | 1.000000 |
| s57e1 | 1.000000 |
| s58e15 | 1.000000 |
| s59e6 | 1.000000 |
| s60e16 | 1.000000 |
| s61e20 | 1.000000 |
| s62e7 | 1.000000 |
| s63e5 | 1.000000 |
| em1w1d1 | 1.000000 |
| em3w1d1 | 1.000000 |
| em4w1d1 | 1.000000 |
| em5w1d1 | 1.000000 |

| | |
|----------|----------|
| em6wld1 | 1.000000 |
| em14wld1 | 1.000000 |
| em16wld1 | 1.000000 |
| em18wld1 | 1.000000 |
| em20wld1 | 1.000000 |
| em1wld2 | 1.000000 |
| em2wld2 | 1.000000 |
| em4wld2 | 1.000000 |
| em7wld2 | 1.000000 |
| em8wld2 | 1.000000 |
| em14wld2 | 1.000000 |
| em17wld2 | 1.000000 |
| em18wld2 | 1.000000 |
| em20wld2 | 1.000000 |
| em1wld3 | 1.000000 |
| em2wld3 | 1.000000 |
| em3wld3 | 1.000000 |
| em4wld3 | 1.000000 |
| em6wld3 | 1.000000 |
| em7wld3 | 1.000000 |
| em9wld3 | 1.000000 |
| em15wld3 | 1.000000 |
| em19wld3 | 1.000000 |
| em3wld4 | 1.000000 |
| em4wld4 | 1.000000 |
| em7wld4 | 1.000000 |
| em9wld4 | 1.000000 |
| em10wld4 | 1.000000 |
| em15wld4 | 1.000000 |
| em16wld4 | 1.000000 |
| em17wld4 | 1.000000 |
| em19wld4 | 1.000000 |
| em2wld5 | 1.000000 |
| em5wld5 | 1.000000 |
| em6wld5 | 1.000000 |
| em8wld5 | 1.000000 |
| em9wld5 | 1.000000 |
| em16wld5 | 1.000000 |
| em17wld5 | 1.000000 |
| em18wld5 | 1.000000 |
| em19wld5 | 1.000000 |
| em2wld6 | 1.000000 |
| em3wld6 | 1.000000 |
| em5wld6 | 1.000000 |
| em8wld6 | 1.000000 |
| em9wld6 | 1.000000 |
| em15wld6 | 1.000000 |
| em17wld6 | 1.000000 |
| em19wld6 | 1.000000 |
| em20wld6 | 1.000000 |
| em1wld7 | 1.000000 |
| em5wld7 | 1.000000 |
| em6wld7 | 1.000000 |
| em7wld7 | 1.000000 |
| em8wld7 | 1.000000 |

| | |
|----------|----------|
| em15wld7 | 1.000000 |
| em16wld7 | 1.000000 |
| em18wld7 | 1.000000 |
| em20wld7 | 1.000000 |

All other variables in the range 1-1034 are zero.